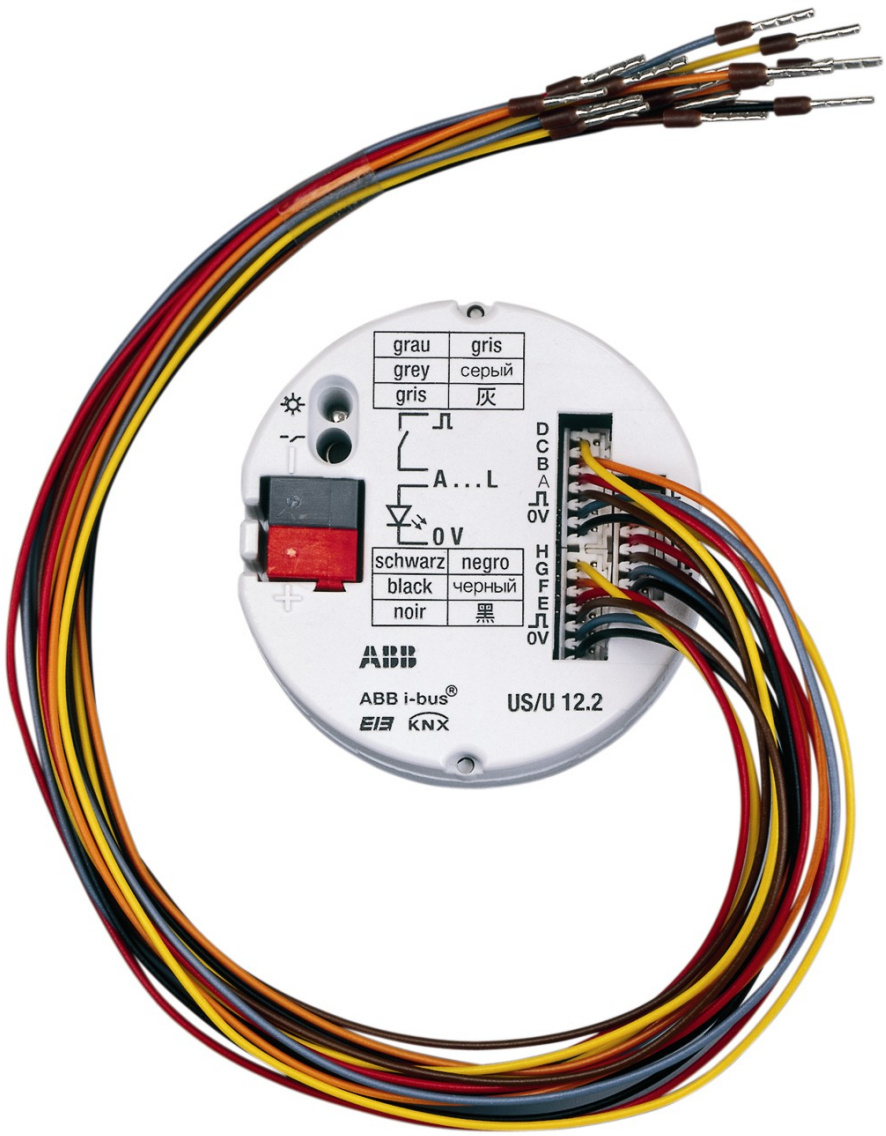
Product Manual

ABB i-bus® EIB / KNX

Universal Interface
US/U 12.2

Intelligent Installation Systems



Contents

This manual describes the function of the Universal Interface US/U 12.2 with the application program “Binary Input Display Heat 12f/1”. Subject to technical changes and errors excepted.

Exclusion of liability:
Despite checking that the contents of this document match the hardware and software, deviations cannot be completely excluded. We therefore cannot accept any liability for this. Any necessary corrections will be inserted in new versions of the manual.
Please inform us of any suggested improvements.

1 General

The Universal Interface US/U 12.2 is used for the operation and display of building functions via push buttons and light emitting diodes (LED's). The compact design enables the device to be inserted in a conventional 60 mm wiring box, e.g. behind an operating panel.

This manual provides technical information about the device as well as its assembly and programming. The last section contains application examples for its effective use on site.

1.1 Product and functional overview

The Universal Interface US/U 12.2 has 12 channels for the connection of conventional push buttons (input mode) or LED's (output mode). Alternatively one Electronic Relay ER/U 1.1 can be connected per channel for controlling electrothermal valve drives.

The operating mode of each channel of the device can be parameterized separately. The connecting lines of 30 cm can be extended up to 10 m.

The supply of the LED's (2 mA per channel) is provided by the device. Therefore no additional power supply is required.

The functionality is extremely extensive but comprehensible and enables the device to be used in a wide variety of application areas. The following list provides an overview:

Switching and dimming of the lighting (also 1 button operation)

- Operation of blinds and shutters (also 1 button operation)
- Sending of values e.g. temperature values
- Control and storing of light scenes
- Triggering an electronic relay for controlling an electrothermal drive mechanism for heating valves
- Triggering an LED (with flashing function and time restriction) for reporting an operation
- Operation of various loads by multiple push button actions
- Operation of several loads in a fixed switching sequence
- Counting of impulses and push button operations

Each channel of a device can adopt any of the functions described above.

2 Device technology

The device has four channels, which can either be parameterized as inputs or outputs in the ETS program.

Using the colour-coded connecting cables, it is possible to connect conventional push buttons, floating contacts or light-emitting diodes.

The scanning voltage for the contacts and the supply voltage for the LED's are made available by the device.

Series resistors for external LED's are integrated in the device.
The universal interface is inserted in a conventional 60 mm combined wall and joint box.

The bus connection is carried out via the bus-connecting terminal supplied.

2.1 Technical data

Supply	- Operating voltage	21...30 V DC, via the bus
	- Current consumption	10 mA
Inputs and Outputs	- Number	12, can be separately parameterized as input or output
	- Permitted cable length	≤ 10 m
Input	- Polling voltage U_n	20 V DC, pulsed
	- Sensing current I_n	0.5 mA
Output	- Output voltage	3.3 V DC
	- Output current	Max. 2 mA
	- Safety	Short circuit proof, overload protection, reverse voltage protection
Operating and display elements	- LED (red) and push button	For assigning the physical address
Connections	- Inputs / Outputs	3x6 cables, approx. 30 cm long, can be extended to max. 10m
	- EIB / KNX	Via bus connecting terminal
Ambient temperature range	- Operation	-5° C ... + 45° C
	- Storage	-25° C ... + 55° C
	- Transport	-25° C ... + 70° C
Type of protection	IP 20 when installed	To EN 60 529
Protection class	III	To DIN EN 61 140
Mounting	In switch box ∅ 60mm	
Mounting position	As required	
Dimensions (∅ x H)	54 x 19 mm	
Weight	0.06 kg	
Housing, colour	Plastic housing, halogen free, colour: grey	
Approvals	EIB / KNX to EN 50 090-1, -2	
CE mark	In accordance with EMC guideline and low voltage guideline	

Application program	Max. number of communication objects	Max. number of group addresses	Max. number of associations
Binary Input Display Heat 12f/1	84	254	255

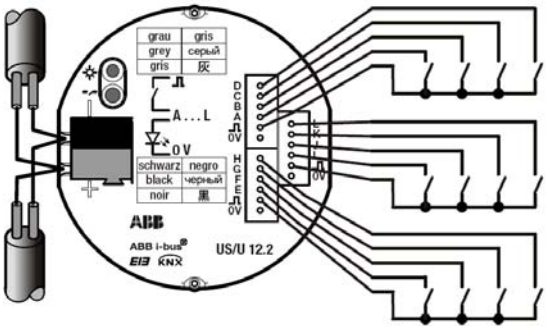
Note: The programming requires EIB Software Tool ETS2 V1.3a or higher. If ETS3 is used a ".VD3" type file must be imported. The application program is available in the ETS2 / ETS3 at "ABB / Display and Visualisation / Input and Output"

Note: The device does not support the encoding function of the ETS. If the access to the device is locked by using a "BC-password" (ETS2) or a "BCU-key" (ETS3) respectively, this will have no effect to this device. It can still be read out or programmed.

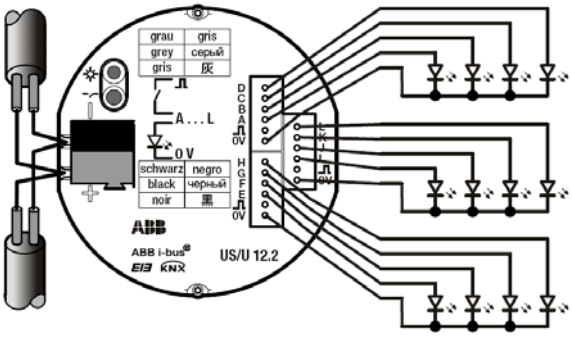
2.2 Circuit diagram

The maximum cable length is 10 m. The colours of the connection cables are explained in section 2.5.

Connection of a floating push button / switch



Connection of an LED



The series resistors for the LED's are integrated in the device. The maximum output current is 2 mA.

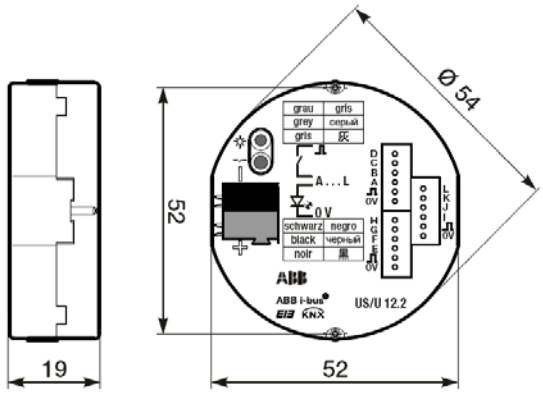
Connection of an Electronic Relay type ER/U 1.1

The electronic relay is connected according to an LED. The coloured core is connected to "+", the black core is connected to "–".

Important: The connection of other relays than type ER/U 1.1 is not allowed.

Note: The connection to an S0 pulse output is possible for electronic energy meters of ABB only. The correct polarity should be observed ("+" to grey core, "– " to coloured core).

2.3 Dimension drawing



2.4 Mounting and installation

The device can in mounted in any position. Any cores that are not required must be insulated.

Accessibility of the devices for the purpose operation, testing, visual inspection, maintenance and repair must be must be provided (conform to DIN VDE 0100-520).

2.5 Description of the inputs and outputs

Grey core (⎍): Positive scanning voltage

During operation as an input, the grey core makes the positive, pulsed scanning voltage available.

Coloured core: Control of the channel

During operation as an input, the status of the contact is read out via the coloured cores.

During operation as an output, the coloured core makes the positive output voltage available.

The following table allocates the colours to the channels:

brown	Channels A, E and I
red	Channels B, F and J
orange	Channels C, G and K
yellow	Channels D, H and L

Black core (0V): Negative reference potential

During operation as an output, the black core makes the negative reference potential available.

Important: The inputs and outputs do not have electrical isolation to the EIB bus voltage (SELV). The SELV criteria only enable the connection of floating contacts with safety separation.

3 Commissioning

3.1 Overview

A powerful application program *Binary Input Display Heat 12f/1* is available for the Universal Interface US/U 12.2. The following functions can be set separately for each input:

Switch sensor	For switching the lighting or scanning a floating contact (relay). Distinction between short/long operation and cyclical sending of the contact state are possible.
Switch/dimming sensor	For switching/dimming the lighting. Start/stop dimming and stepwise dimming, as well as dimming via a single push button are possible.
Shutter sensor	For movement/louvre adjustment of a shutter or blind. Eight present operation modes are possible in total.
Value / Forced operation	For sending the values of different data types (e.g. temperature values). It is possible to send different values or data types for short/long operation, possible to activate/deactivate the forced operation of actuators.
Control scene	For recalling and storing the states of several actuator groups. The actuator groups can either be controlled via max. 5 individual objects or (if supported by the actuators) via a special 8 bit scene object.
Control valve drive	For controlling an electrothermal valve drive. The device controls an Electronic Relay type ER/U 1.1 to which an electrothermal valve drive can be connected. The device has the full functionality of a heating actuator. Control via 2-step controller or continuous controller (PWM), cyclical valve purging, monitoring of the room thermostat and forced operation of the valve drive are possible.
Control LED	For controlling a light-emitting diode. Switching and flashing (with time limit and various flashing rates) and use as an orientation light are possible.
Switching sequence	For the operation of several actuators in a succession. The actuators are switched in a given sequence that can be selected.
Multiple operation	For triggering various functions depending on the number of operations. Using this function, for example a double operation can switch off alls the lighting in a room. A long operation can also be detected and a function can be triggered.

Supplied state

The Binary Input is supplied with the physical address 15.15.255. The application program is preinstalled. The entire application can be reloaded as required by previously unloading the device.

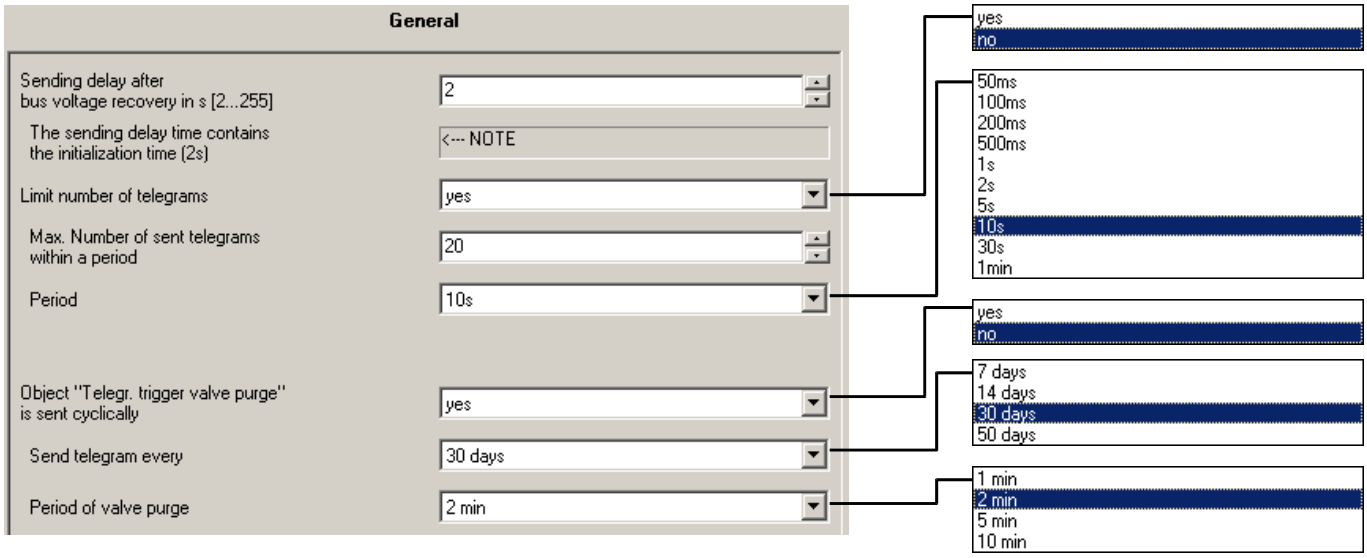
A longer downtime may result if the application program is changed or after a discharge.

3.2 Parameters and communication objects

3.2.1 General Parameters

Parameters for the functions which affect the complete device can be set via the *General* parameter window.

3.2.1.1 Parameter window: "General"



Sending delay after bus voltage recovery in s [2...255]

The sending delay determines the period between bus voltage recovery and the point after which telegrams can be sent. An initialisation period of approx. 2 seconds for starting the device is included in the sending delay.

If objects are read out via the bus during the sending delay (e.g. from visualisation terminals), these requests are stored and are answered once the sending delay has elapsed.

How does the device behave after bus voltage recovery?

After bus voltage recovery, there is a transmission delay before telegrams are sent on the bus. The following diagram indicates the time sequence:

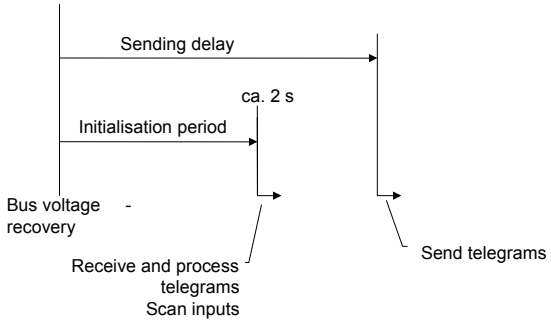


Diagram 1: Behaviour after bus voltage recovery

On bus voltage recovery, the inputs are scanned after the initialisation period and the object values are update accordingly, if possible. If the input is operated, the device behaves as if the operation had started at the end of the initialisation period.

Special behaviour of the operating modes

The behaviour is dependent on the operating mode of the channel. The following list provides an overview:

Operating mode	Behaviour on bus voltage recovery
Switch sensor	If there is a distinction between a short and a long operation or the value *TOGGLE* has been set in one of the parameters *Reaction on closing/opening the contact*, no telegrams are sent after bus voltage recovery. Otherwise the behaviour can be set in the parameters.
Switch/dimming sensor	No telegrams are sent on the bus.
Shutter sensor	No telegrams are sent on the bus.
Value / forced operation	Object values are overwritten with the parameterised values.
Control scene	When the scene is controlled via “5 separate objects”, the object values of the scene are overwritten with the parameterised values.
Control valve drive	Until the first telegram of the room thermostat has been received, the parameterised value is set.
Control LED	The status of the output can be set in the parameters.
Switching sequence	No telegrams are sent on the bus.
Multiple operation	No telegrams are sent on the bus.

Limit number of telegrams

In order to check the bus load which is generated by the device, there is a powerful limit function for telegrams. It is possible to set how many telegrams (***Max. number of telegrams within a period***) can be sent within an adjustable ***period***.

How does the limit function for telegrams work?

A new monitoring period starts at the end of the previous monitoring period. The sent telegrams are counted. As soon as the *Max. number of sent telegrams within a period* has been reached, no further telegrams are sent on the bus until the end of the monitoring period.

When a new monitoring period starts, the telegram counter is reset to zero and the sending of telegrams is again permitted.

Object ‘Telegr. Trigger valve purge’ is sent cyclically

This function is only relevant if one or several channels are used to control a valve drive. Regular purging of a heating control valve can prevent deposits from building up in the valve thereby restricting the valve function. This is particularly significant during periods when only a few changes are made to the valve position.

If this parameter is set to *yes*, the object *Telegr. trigger valve purge* is visible. It is sent at adjustable intervals to start the valve purge (***Send telegram every***) and has the value “1” for the ***Period of valve purge***. The *Valve purge* object of a channel which has been assigned the function of a heating actuator can be controlled via this object.

3.2.1.2 General communication objects

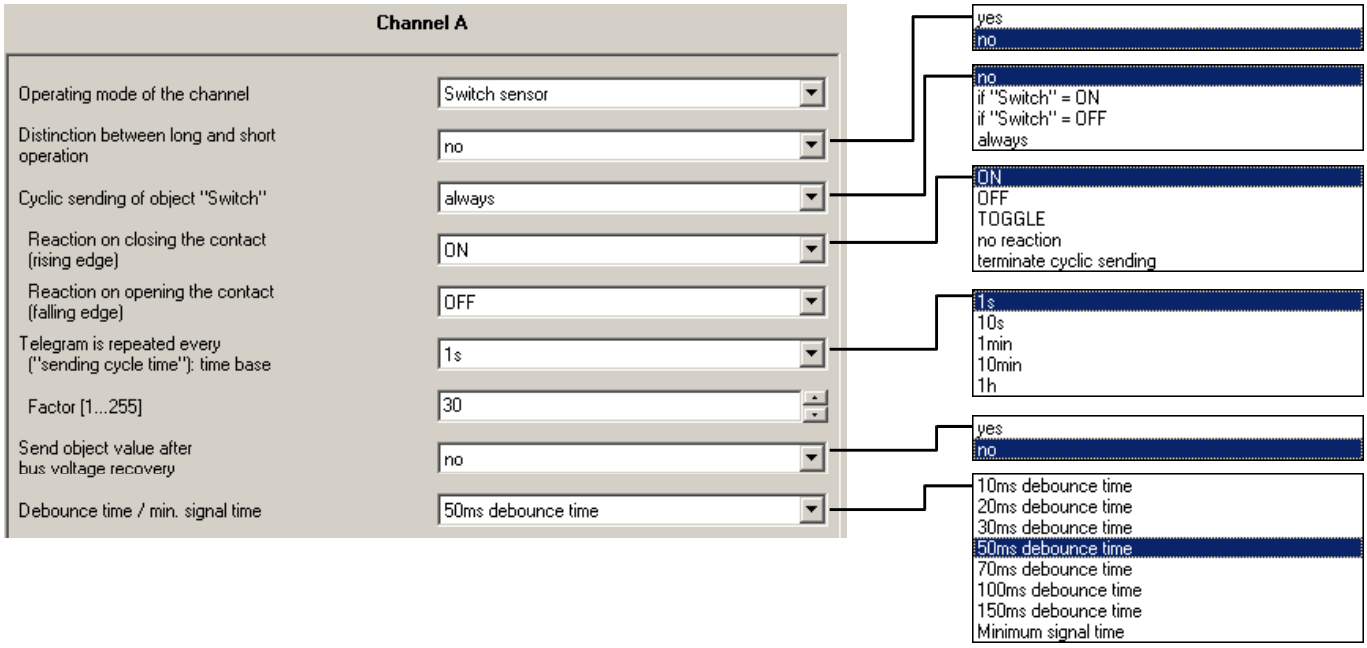
No.	Function	Object name	Data type	Flags
85	**Telegr. trigger valve purge**	**General**	**1 Bit EIS1 DPT 1.001**	**CT**
The object is set at regular intervals to the value “1” for an adjustable period and then reset to “0”. It can be used to trigger a valve purge at regular intervals (see *Valve purge* object). After bus voltage recovery, this object sends the value “0” to the bus and the purge cycle is restarted. This object is visible if the parameter *Send object ‘Telegr. valve purge’* is set to *yes*.				

3.2.2 Operating mode: “Switch sensor“

The “switch sensor operating mode” is described in the following.

3.2.2.1 Parameters “without short/long operation”

The following parameters are visible, if the parameter *distinction between long and short operation* has the value *no*.



Distinction between long and short operation

If the parameter is set to *no* the input will be evaluated normally on every edge of the input signal.

If *yes* is selected, there is a delay after opening/closing the contact to determine whether there is a short or long operation. Only then is a possible reaction triggered.

The following diagram illustrates the function:

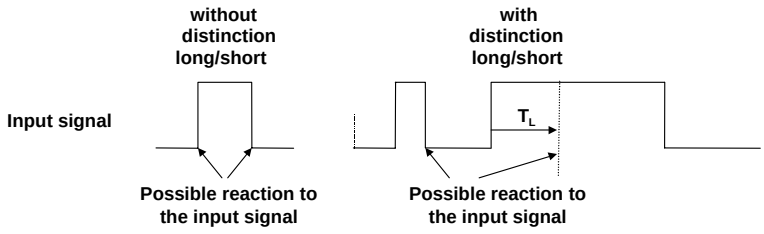


Diagram 2: Distinction between short/long operation for “Switch sensor” function

T_L is the period after which a push button action is recognised as a long operation.

Cyclic sending of object “Switch”

This parameter is visible if there is no distinction between a short and long operation. It defines whether the object “Switch” is sent cyclically on the bus.

option *always*: the object sends cyclically on the bus, regardless of its value.

option *if ‘switch’ = ON*: only the value “1” is sent cyclically.

option *if ‘switch’ = OFF*: only the value “0” is sent cyclically.

The cyclical sending of object *Switch* can be used e.g. to monitor the life signs of the sensor.

How does the cyclic sending work?

Cyclic sending enables the communication object “Switch” to send automatically at a fixed interval.

If cyclical sending is only carried out for a specific object value (ON or OFF), this condition refers to the value of the communication object. It is therefore possible in principle to start the cyclical sending by sending a value to the communication object *Switch*. As this reaction is generally unwanted, the “write” flag and “update” flag of the communication object have to be deleted in the setting to ensure that it cannot by changed via the bus. If however this functionality is required, the flags must be set accordingly.

When the “Switch” object changes and after bus voltage recovery, the object value is sent immediately on the bus and the sending cycle time restarts.

Reaction on closing the contact
Reaction on opening the contact

This parameter is visible if there is no distinction between a short and long operation. It can be set separately for each pulse edge whether the object value should be “ON, “OFF”, “TOGGLE” or “no reaction”.

If cyclical sending has been parameterised, it is possible to select the option *terminate cyclic sending* so that an operation of the input can stop the cyclical sending without a new object value being sent.

Sending cycle time: Telegram is repeated every

This parameter is visible if cyclical sending has been set. It describes the interval between two telegrams that are sent cyclically:

Sending cycle time = Time Base x Factor.

Send object value after bus voltage recovery

It can be set whether the current status of the input is sent on the bus via object *Switch* after bus voltage recovery (once the sending delay has elapsed).

A value is however only sent on the bus if the value *TOGGLE* has not been set in either of the two parameters *Reaction on opening/closing the contact*.

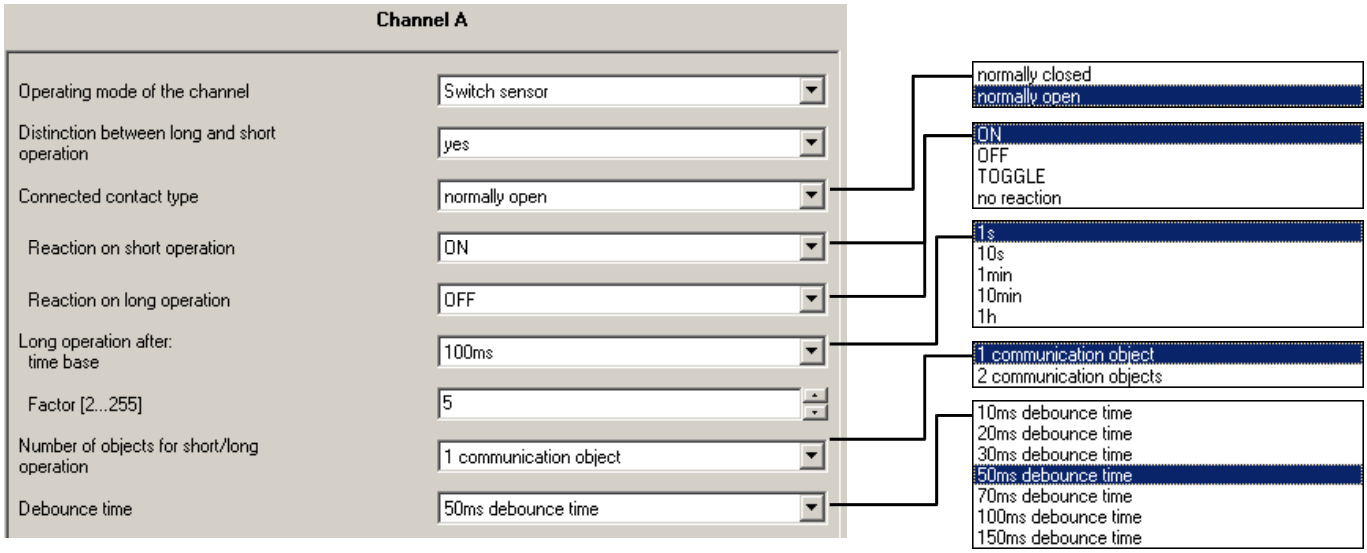
If one of the two parameters has the value *TOGGLE*, no values are sent in general on the bus after bus voltage recovery.

Debounce time / min. signal time

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 4.1 for the exact function of this parameter.

3.2.2.2 Parameters “with short/long operation”

The following parameters are visible, if the parameter *distinction between long and short operation* has the value *yes*.



Distinction between long and short operation

If the parameter is set to *no* the input will be evaluated normally on every edge of the input signal.

If *yes* is selected, there is a delay after opening/closing the contact to determine whether there is a short or long operation. Only then is a possible reaction triggered.

The following diagram illustrates the function:

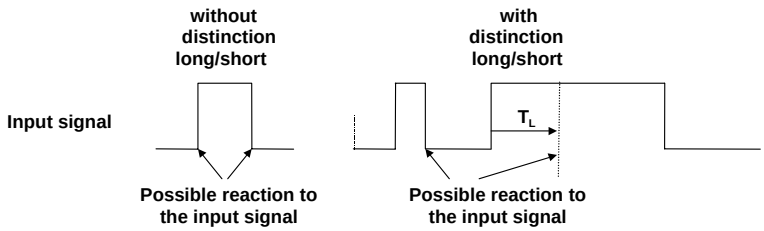


Diagram 3: Distinction between short/long operation for “Switch sensor” function

T_L is the period after which a push button action is recognised as a long operation.

Connected contact type

normally open: The input is closed in state of operation.
normally closed: The input is open in state of operation.

Reaction on short operation
Reaction on long operation

It can be set for each operation at the input (short or long) how the object value is changed. The object value is updated as soon as it is established whether the operation is long or short.

Long operation after: time base, factor

This parameter is visible if there is a distinction between a short and long operation. The period T_L is defined here, after which an operation is interpreted as “long”.

T_L = Time Base x Factor

Number of objects for short/long operation

To differentiate between short and long operations, it is possible to activate a further object by setting the parameter value *2 communication objects*. This object only reacts on long operations, while the existing object only reacts on short operations.

Debounce time

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 4.1 for the exact function of this parameter.

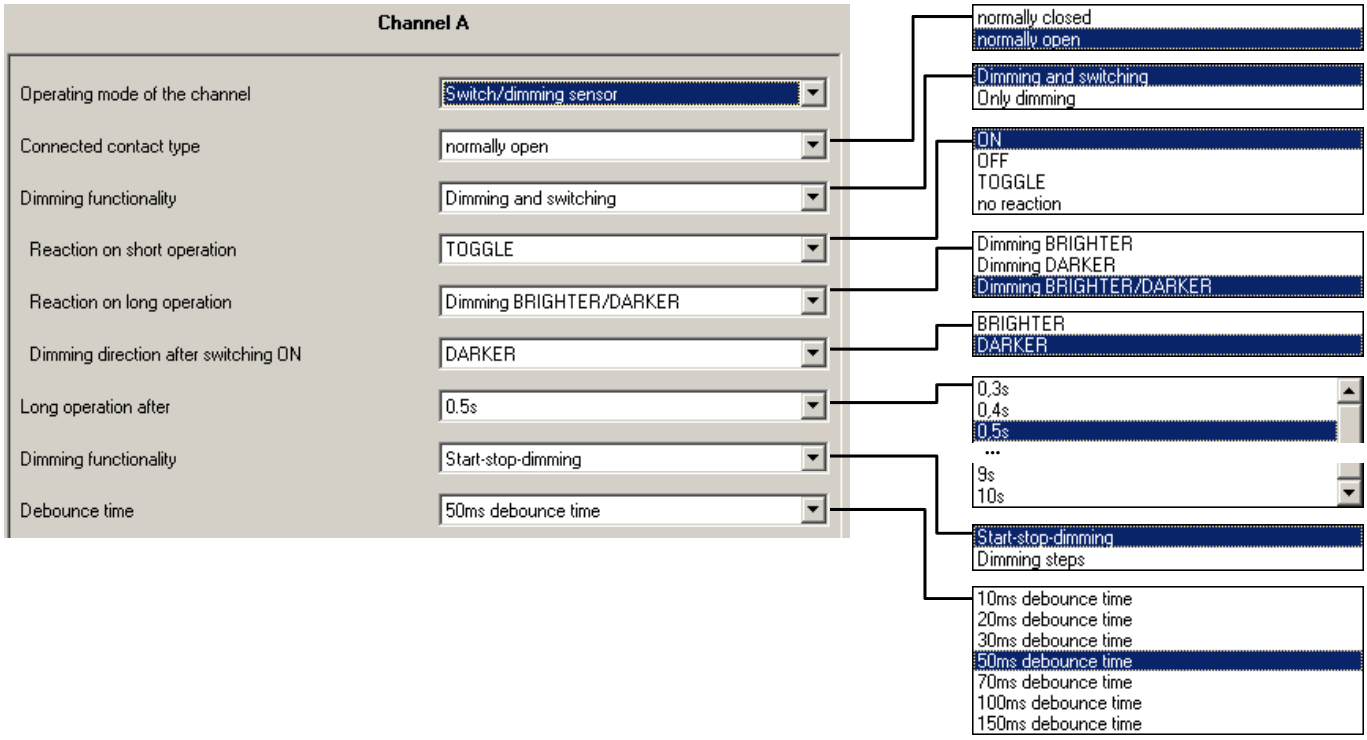
3.2.2.3 Communication objects “Switch sensor”

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 Bit EIS1 DPT 1.003**	**CW**
0: Release channel 1: Disable channel The channel circuitry can be blocked or released via this communication object. A blocked channel behaves as if the input signal does not occur. The communication objects of the channel continue to be available. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the status of the input has changed during the blocking. If the channel is just being operated as it is being enabled, the channel behaves as if the operation has just commenced. The behaviour of the channel is undefined if the channel is blocked during operation.				
1	**Switch**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CWT**
0: OFF 1: ON In accordance with the parameter setting, this communication object can be switched by actuation of the ON ,OFF or TOGGLE input. With TOGGLE the previous value e.g. “1” is switched directly to the value “0”.				
2	**Switch**	**Channel A, long operation**	**1 Bit EIS1 DPT 1.001**	**CWT**
0: OFF 1: ON This communication object is only visible if the parameter *Distinction between long and short operation = yes*, and the parameter *Number of objects for short/long operation = 2 communication objects*. This additional communication object is assigned to the long operation. If this object is visible, then the existing communication object *Switch* only reacts to a short operation.				

3.2.3 Operating mode: “Switch/dimming sensor”

The “switch/dimming sensor operating mode” is described in the following.

3.2.3.1 Parameters



Connected contact type

normally open: The input is closed in state of operation.
normally closed: The input is open in state of operation.

Dimming functionality

This parameter determines whether the lighting is only dimmed (*Only dimming*) or whether it should also be switched (*Dimming and switching*). In this case, the lighting is dimmed via a long operation and switched via a short operation.

The benefit of the setting *Only dimming* is that there is no distinction between a short and long operation. The dimming command is therefore carried out immediately after the push button action; there is no delay to determine whether the operation is long or short.

How does “1 button dimming” function?

Switching and dimming functions can be fully controlled via a single push button. Each dim actuation is sent alternately with a BRIGHTER or DARKER dim telegram.

The **1 button dimming** is preset in the parameters. The function is the following: If the communication object “Switch” = 0, a BRIGHTER telegram is sent at all times in case of a long operation. In order to evaluate the switch feedback of the actuator, the “Write” flag of the communication object “Switch” is set.

The following table illustrates the function in detail:

Value of the object “Switch”	Value of the last dimming telegram	Reaction to dimming operation (sent dimming telegram)
OFF	DARKER	BRIGHTER
OFF	BRIGHTER	BRIGHTER

ON	DARKER	BRIGHTER
OFF	BRIGHTER	DARKER

Table 1 : Dimming function „1 button dimming“

How does “2 button dimming” function?

2 button dimming requires 2 channels, e.g. Channel A with short operation for switch on and long operation for dim brighter. Channel B with short operation for switch off and long operation for dim darker.

The objects “Switch” and “Dimming” of both channels have to be assigned to the same group addresses.

The user thus has complete freedom to choose which push buttons are combined with one another in order to dim a group of luminaries, or which function the individual push button has in this case.

Reaction on short operation

This parameter is visible if the value *Dimming and switching* has been set in the parameter *Dimming functionality*.

A short operation changes the value of the object *Switch*. This parameter sets whether the object *Switch* is toggled after a short operation (typically: dimming with 1 button) or only switched on or off (typically: dimming with 2 buttons).

Reaction on long operation

This parameter is visible if the value *Dimming and switching* has been set in the parameter *Dimming function*.

A long operation changes the value of the object *Dimming*. This parameter sets whether the object Dimming sends a dim brighter or a dim darker telegram after a long operation. The setting “Dim BRIGHTER/DARKER” must be selected for dimming with 1 button. The opposite dimming command to the last command is sent in this case.

Dimming direction after dimming ON

In this parameter can be set whether the lighting should dim brighter or darker after switching on.

Example: If the brightness on switching on is 20% only, it makes sense to dim brighter after the lighting was switched on.

Long operation after

This parameter is visible if the value *Dimming and switching* has been set in the parameter *Dimming functionality*. The period T_L is defined here, after which an operation is interpreted as “long”.

Reaction on operation

This parameter is visible if the value *Only dimming* has been set in the parameter *Dimming functionality* (no distinction between a short and long operation). The meaning of the parameter settings corresponds to those of the parameter “Reaction on long operation” (see above).

Dimming mode

Start-stop-dimming is the normal dimming mode. It begins the dimming process with a dim darker or brighter telegram and ends the dimming process with a stop telegram. Cyclical sending of the dimming telegram is not required in this case.

For *Dimming steps*, the dimming telegram is sent cyclically during a long operation. Once the operation has finished, a stop telegram ends the dimming process.

Brightness change on every sent telegram

This parameter is only visible for *Dimming steps*. It can be set, which change in brightness (percentage value) is caused by each cyclically sent dimming telegram.

Telegram is repeated every (“sending cycle time”)

If *Dimming steps* has been set, the dimming telegram is sent cyclically during a long operation. The transmission cycle time corresponds to the interval between two telegrams during cyclical sending.

Debounce time

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 4.1 for the exact function of this parameter.

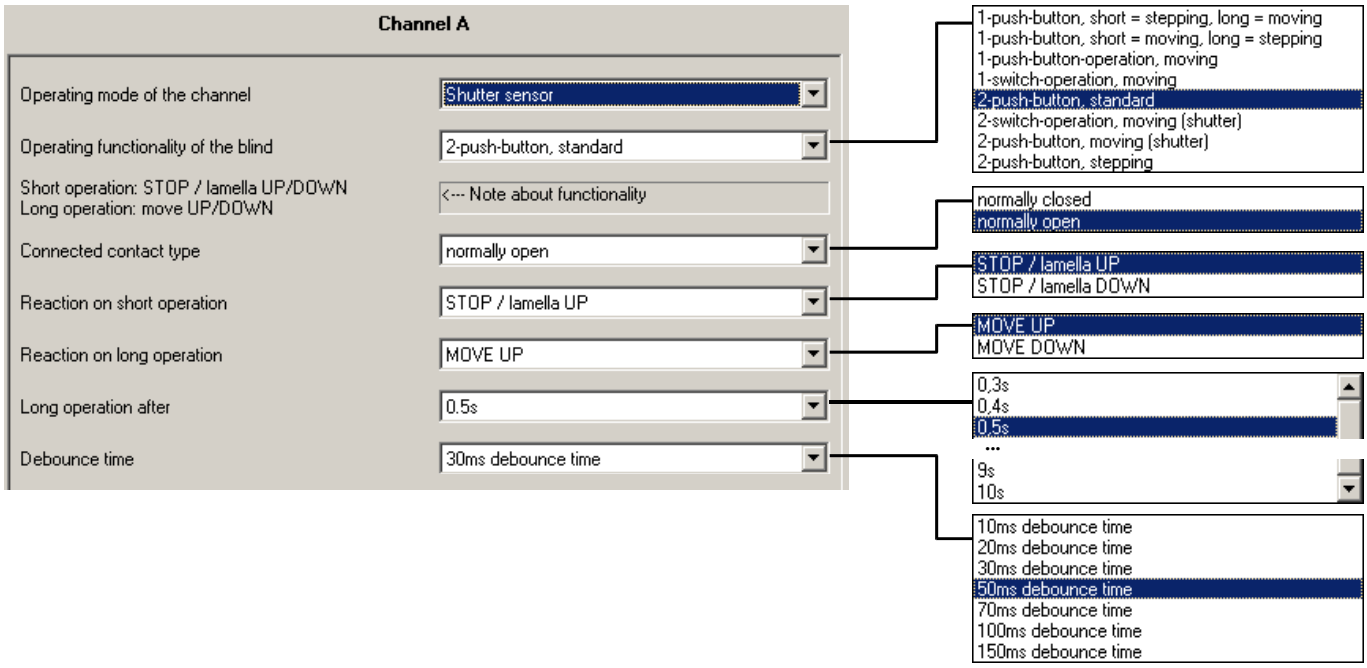
3.2.3.2 Communication objects “Switch/dimming sensor”

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 Bit EIS1 DPT 1.003**	**CW**
0: Release channel 1: Disable channel The channel circuitry can be blocked or released via this communication object. A blocked channel behaves as if the input signal does not occur. The communication objects of the channel continue to be available. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the status of the input has changed during the blocking. If the channel is just being operated as it is being enabled, the channel behaves as if the operation has just commenced. The behaviour of the channel is undefined if the channel is blocked during operation.				
1	**Switch**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CWT**
This object is visible if the value *Switching and dimming* has been set in the parameter *Dimming functionality*. Depending on the parameter setting, the object value can be switched on, off or toggled after a short operation. For dimming with 1 button, this object should be linked with the status response of the dimming actuator as a nonsending group address. The input is thus informed about the current switching status of the dimming actuator.				
2	**Dimming**	**Channel A**	**4 Bit EIS2 DPT 3.007**	**CT**
A long operation of the input causes a dim brighter or dim darker command to be sent on the bus via this object. A stop command is sent at the end of this operation.				

3.2.4 Operating mode: ”Shutter sensor“

The “shutter sensor operating mode” is described in the following. The function enables the operation of blinds and shutters with push buttons or switches.

3.2.4.1 Parameters



Operating functionality of the blind

The following tables provide and overview of the shutter operating functions:

1 push button, (short = stepping, long = moving)	
Short operation	Stop/lamella adjustment; Opposite direction to the last movement command To return to lamella adjustment, the blind must be raised or lowered briefly.
Long operation	Alternately *MOVE UP* or *MOVE DOWN*
1 push button, short = moving, long = stepping	
Short operation	Alternately *MOVE UP* or *MOVE DOWN*
Long operation	STOP/lamella adjustment (cyclical sending); Opposite direction to the last movement or stepping command
1 push button, moving / stopping only	
On operation	The following commands are sent in sequence: *... MOVE UP → STOP / lamella UP —>* *MOVE DOWN → STOP / lamella DOWN → ...*
1 switch, moving only	
Start of operation	Alternately *MOVE UP* or *MOVE DOWN*
End of operation	STOP / lamella adj.
2 push button, standard (short = stepping, long = moving)	
Short operation	*STOP/lamella UP* or *... DOWN* (programmable)
Long operation	*MOVE UP* or *MOVE DOWN* (programmable)

2 switches, moving only	
Start of operation	*MOVE UP* or *MOVE DOWN* (programmable)
End of operation	*STOP/lamella UP* or *... DOWN* (programmable)
2 push buttons, moving / stopping only	
On operation	The following commands are sent in sequence: ... → *MOVE UP* → *STOP / lamella UP* → ... or ... → *MOVE DOWN* → *STOP / lamella UP* → …
2 push buttons, louvre adjustment only	
On operation	*STOP / lamella adj. UP* or *... DOWN*

Connected contact type

normally open: The input is closed in state of operation.
normally closed: The input is open in state of operation.

Reaction on operation

This parameter is visible in all operating modes in which there is no distinction between a short and long operation. It can be set whether the input triggers commands for upward movement (*UP*) or downward movement (*DOWN*).

Reaction on short operation
Reaction on long operation

This parameter is visible in all operating modes in which there is a distinction between a short and long operation. It can be set whether the input triggers commands for upward movement (*UP*) or downward movement (*DOWN*).

Long operation after

This parameter is visible in all operating modes in which there is a distinction between a short and long operation. The period which defines a long operation is set here.

“STOP/Lamella adjustment” is repeated every

This parameter is visible in operating modes in which the object *STOP/lamella adjustment* is sent cyclically on the bus during a long operation. The interval between two telegrams is set here.

Debounce time

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 4.1 for the exact function of this parameter.

3.2.4.2 Communication objects “Shutter sensor”

No.	Function	Object name	Data type	Flags

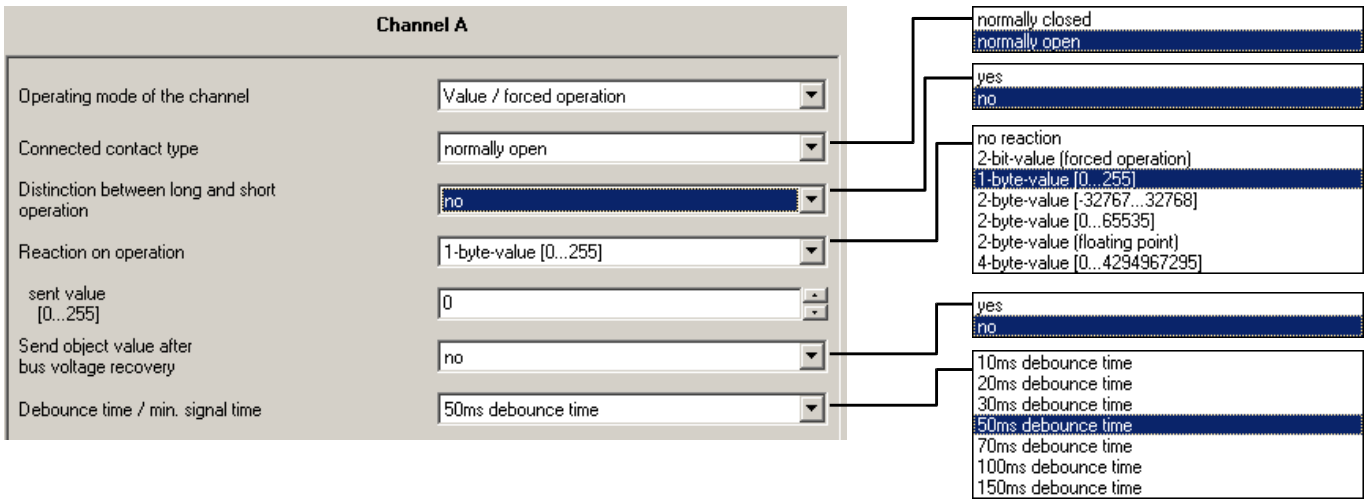
No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 Bit EIS1 DPT 1.003**	**CW**
0: Release channel 1: Disable channel The channel circuitry can be blocked or released via this communication object. A blocked channel behaves as if the input signal does not occur. The communication objects of the channel continue to be available. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the status of the input has changed during the blocking. If the channel is just being operated as it is being enabled, the channel behaves as if the operation has just commenced. The behaviour of the channel is undefined if the channel is blocked during operation.				
1	**Shutter UP/DOWN**	**Channel A**	**1 Bit EIS7 DPT 1.008**	**CWT**
0: move upwards (UP) 1: move downwards (DOWN) This communication object sends a shutter motion command (UP or DOWN) to the bus. The device also detects movement commands of other sensors when telegrams are received (e.g. parallel operation).				
2	**STOP / lamella adjustment**	**Channel A**	**1 Bit EIS7 DPT 1.007**	**CT**
0: STOP / lamella adjustment UP 1: STOP / lamella adjustment DOWN This communication object sends a stop command or lamella adjustment				
3	**Upper limit position**	**Channel A**	**1 Bit EIS1 DPT 1.002**	**CW**
0: no upper end limit 1: at upper end limit Via this communication object, the shutter actuator reports whether or not it is in the upper limit position (“shutter open”). **Note:** The communication object is important for 1 button operation.				
4	**Lower limit position**	**Channel A**	**1 Bit EIS1 DPT 1.002**	**CW**
0: no upper end limit 1: at upper end limit Via this communication object, the shutter actuator reports whether or not it is in the lower limit position (“shutter closed”). **Note:** The communication object is important for 1 button operation.				

3.2.5 Operating mode: “Value / forced operation”

The “value / forced operation operating mode” is described in the following. It allows the sending of various data types.

3.2.5.1 Parameters “without long/short operation”

The following parameters are visible, if the parameter *distinction between long and short operation* has the value *no*.



Connected contact type

normally open: The input is closed in state of operation.
normally closed: The input is open in state of operation.

Distinction between long and short operation

This parameter defines whether the input distinguishes between a short and long operation.

If *yes* is selected, there is a delay after opening/closing the contact to determine whether there is a short or long operation. Only then is a possible reaction triggered.

In the following those parameters are described, which are visible, if no distinction between long and short operation is selected.

Reaction on operation

This parameter defines the data type that is sent when the contact is pressed.

Sent value

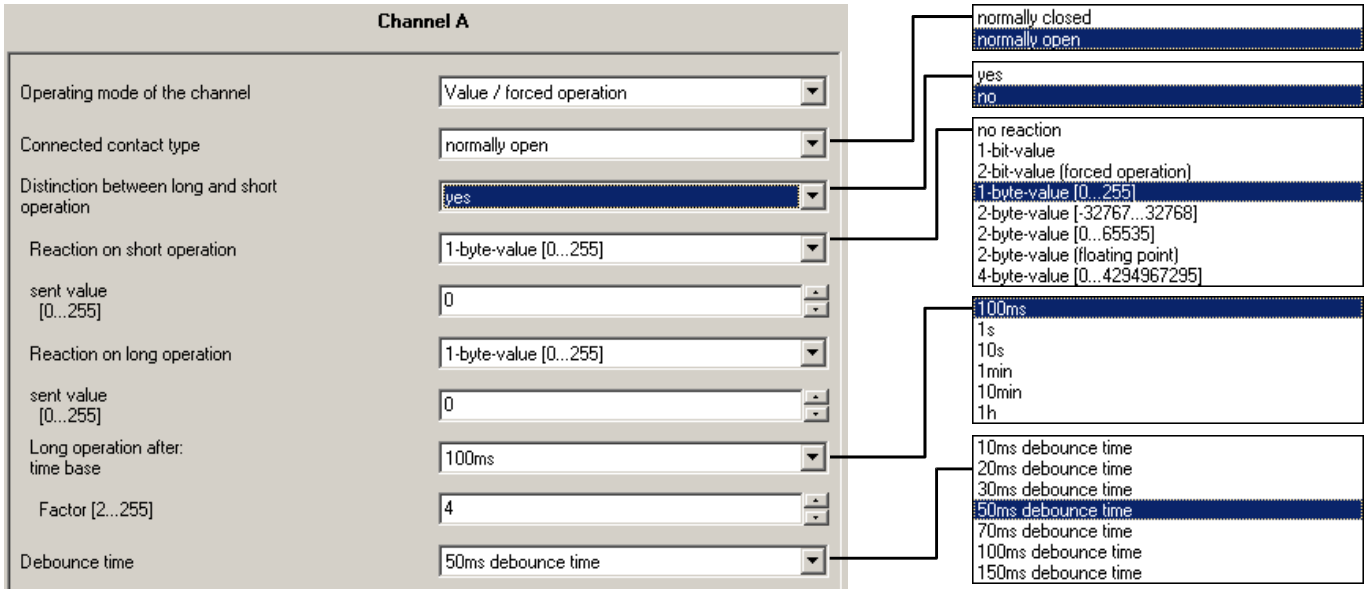
This parameter defines the value which is sent on operation. The value range is dependent on the selected data type.

Debounce time / min. signal time

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 4.1 for the exact function of this parameter.

3.2.5.2 Parameters “with long/short operation”

The following parameters are visible, if the parameter *distinction between long and short operation* has the value *yes*.



Connected contact type

normally open: The input is closed in state of operation.
normally closed: The input is open in state of operation.

Distinction between long and short operation

This parameter defines whether the input distinguishes between a short and long operation.

If *yes* is selected, there is a delay after opening/closing the contact to determine whether there is a short or long operation. Only then is a possible reaction triggered.

In the following those parameters are described, which are visible, if a distinction between long and short operation is selected.

Reaction on short operation
Reaction on long operation

This parameter is visible if there is no distinction between a short and long operation. It defines the data type that is sent after a short or long operation.

Sent value

This parameter defines the value which is sent on operation. The value range is dependent on the selected data type. Two values can be set here when there is a distinction between a short and long operation.

Long operation after: time base, factor

Defines the period T_L after which an operation is interpreted as “long”.

(T_L = time base x factor)

Debounce time

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 4.1 for the exact function of this parameter.

3.2.5.3 Communication objects “Value / forced operation”

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 Bit EIS1 DPT 1.003**	**CW**

0: Release channel
1: Disable channel

The channel circuitry can be blocked or released via this communication object. A blocked channel behaves as if the input signal does not occur. The communication objects of the channel continue to be available.

When a disabled input is enabled, no telegrams are initially sent on the bus, even if the status of the input has changed during the blocking. If the channel is just being operated as it is being enabled, the channel behaves as if the operation has just commenced. The behaviour of the channel is undefined if the channel is blocked during operation.

1	**Value (...)**	**Channel A**	**EIS variable DPT variable**	**CT**

This communication object sends a value to the bus. The value and data type can be freely set in the parameters.

1 Bit [0 / 1]	EIS 1	DPT 1.001	switch command
2 Bit [0...3]	EIS 8	DPT 2.001	forced operation
1 Byte [0...255]	EIS 6	DPT 5.010	brightness or position
2 Byte [-32768...+32767]	EIS 10	DPT 7.001	signed value
2 Byte [0...65535]	EIS 10	DPT 8.001	unsigned value
2 Byte [floating point value*]	EIS 5	DPT 9.001	temperature
4 Byte [0...4294967295]	EIS 11	DPT 12.001	unsigned value

*sends values with the firm exponent of 3

1 **2**	**Value (...)** **Value (...)**	**Channel A, short operation** **Channel A, long operation**	**EIS variable DPT variable**	**CT**

These communication objects send a value to the bus in case of long or short operation respectively. The values and data types can be selected in the parameters (see above).

Note: As standard the “Write” flag with the value objects (except for 1-bit objects) is deleted. Thus, the object value can not be modified via the bus. If this function is required, the “Write” flag must be set in the ETS. The object value is overwritten with the parameterised value on bus voltage recovery.

3.2.6 Operating mode: ”Control scene“

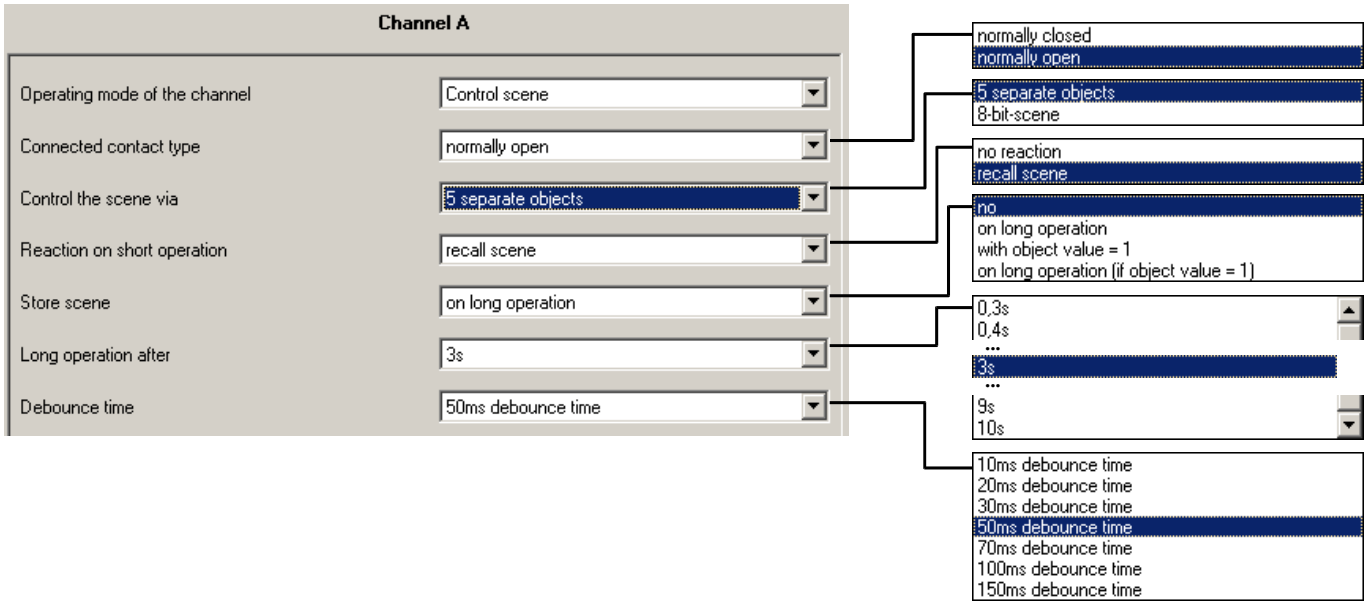
The “control scene operating mode” is described in the following. A scene can be controlled via *5 separate objects* or an *8-bit-scene*.

3.2.6.1 Parameters for control via *5 separate objects*

This operating mode enables the recall and saving of actuator groups. An actuator group comprises several communication objects that are linked with the same group address.

It can consist of e.g. switch actuators (1-bit values) or dimming actuators (1-byte values).

This parameter window becomes visible, if a scene is controlled via *5 separate objects*.



Connected contact type

normally open: The input is closed in state of operation.
normally closed: The input is open in state of operation.

Control the scene via

It is possible to select whether the scene control is carried out via *5 separate objects* or whether values that are stored in the actuators are recalled and saved via an *8 bit scene*.

How does a scene control via *5 separate objects* function?

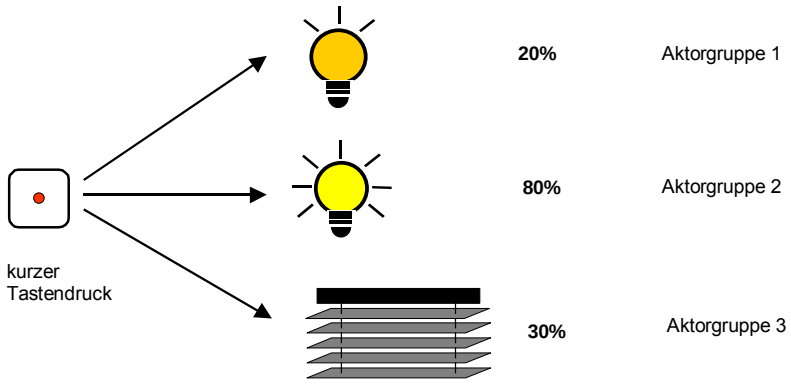


Diagram 1: Calling scene via *5 separate objects*

One input recalls up to 5 actuator groups. Up to 5 telegrams can thus be sent.

An actuator group comprises several actuators that are linked with the same group address. It can consist of e.g. switch actuators (1-bit values) or dimming actuators (1-byte values).

How is a scene stored?

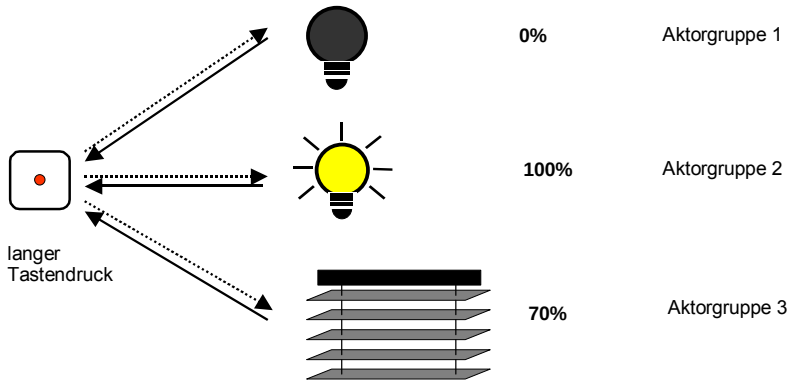


Diagram 2: Storing scene via 5 separate objects

The storing of a scene is carried out with a long push button action. The device scans each individual actuator group for the current value and then stores this value as the new scene value.

Please note that the “Read” flag has to be set for at least one actuator in each actuator group.

How does the user operate a scene?

The operation is very simple: A short operation calls a scene. A long operation stores a scene.

What happens on bus voltage failure?

On bus voltage failure the scenes are reset to the parameterised values. Scenes that have been stored by the user will be lost.

Reaction on short operation

This parameter indicates whether a lightscene is recalled after a short operation of the input or whether no reaction should take place.

Store scene

This parameter defines in which way the storing of the current scene can be triggered as well as the functionality of the object “Store scene”. This is dependent on the scene control. The following table provides an overview:

Parameter value	Behaviour
“on long operation”	On a long keypress, the object values *Actuator group …* are read out via the bus and stored in the object values. During the long operation, it is still possible to modify the objects “Switch/Telegr. value actuator group A..E” via the bus. At the end of the long operation, the object “Store scene” sends the value “0” on the bus and the current object values are stored in the device. At the same time, the object “Store scene” sends the value “1” on the bus. The object value is reset to “0” at the end of the keypress. It can thus be used to signal a successful storage.
“if object value = 1“	If the object *Store scene* receives the value “1”, the object values *Actuator group A..E* are read out via the bus. On receipt of the object value “0”, the current object values are stored in the device. **Important:** **The storing of a scene thus requires the object values “1” and “0” to be sent in succession.**
“on long operation (if object value = 1)”	If the object *Store scene* receives the value “1” on the bus, the next long push button action leads to the scanning of the object values *Actuator group A..E* then takes place via the bus. These values are stored as the new scene values. At the same time, the object “Store scene” sends the value “1” on the bus. The object value is reset to “0” at the end of the keypress. It can thus be used to signal a successful storage. A long operation will be interpreted like a short operation if, the object value *Store scene* has the value “0”. Normally, the scene will thus be called.

Long operation after

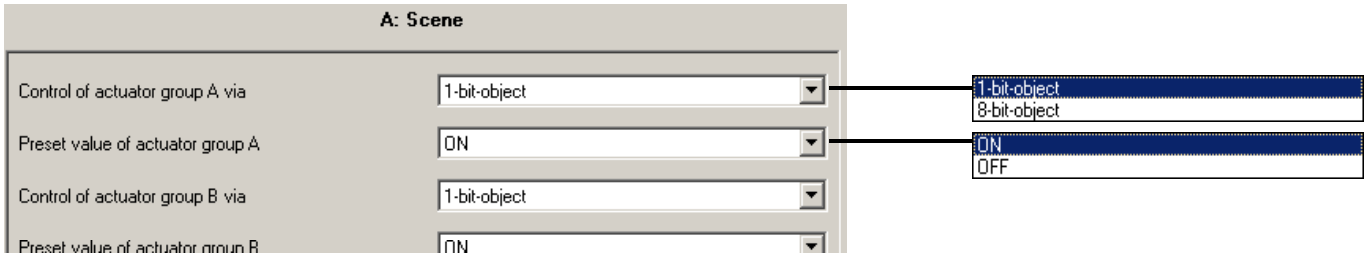
This parameter is visible if the storing of the scene is possible via a long operation. The period after which an operation is interpreted as “long” is defined here.

Debounce time

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 4.1 for the exact function of this parameter.

Parameter window „A: Scene“

This parameter window is visible if a light scene control via *5 separate objects* is selected:



Control actuator group A via
...
Control actuator group E via

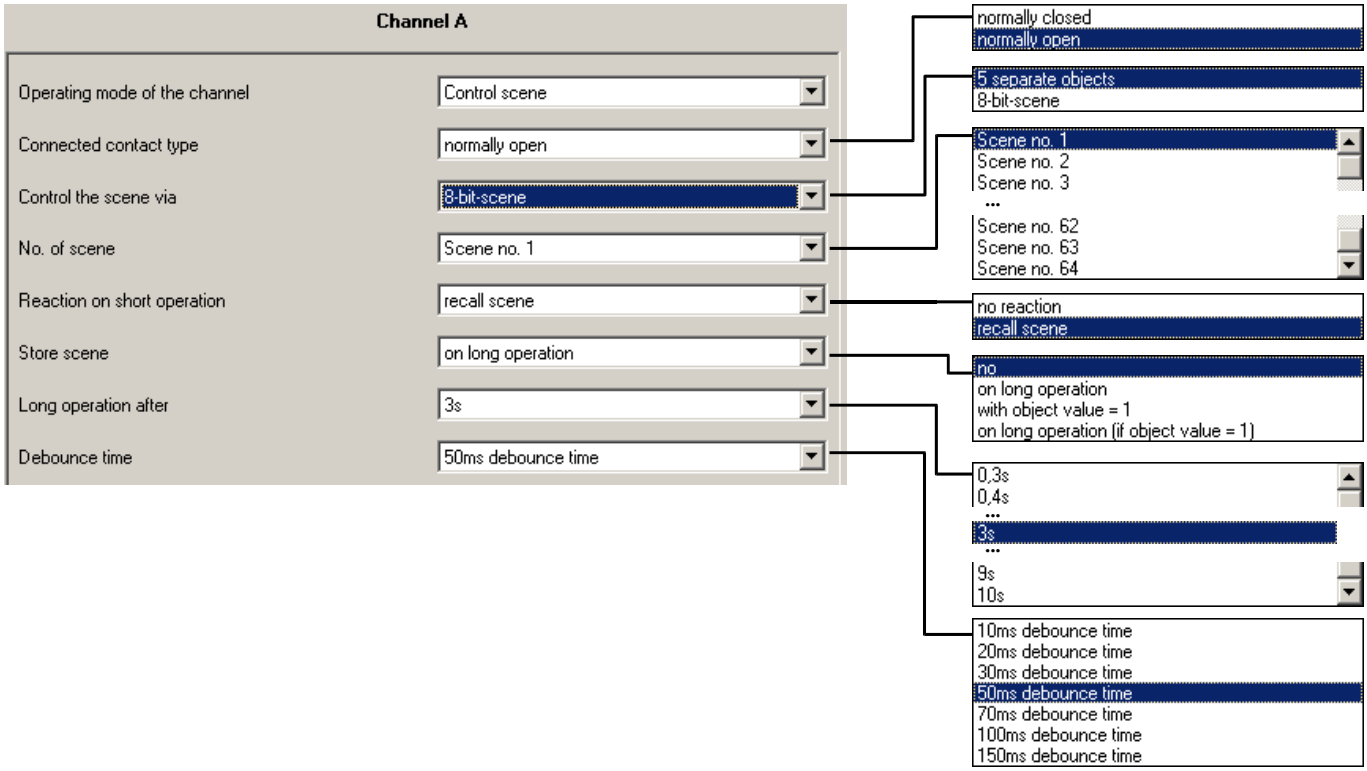
It can be set for each actuator group whether the control is carried out via a *1 bit object* or an *8 bit object*. The type of the communication object *Actuator group…* is set accordingly.

Preset value of actuator group A
...
Preset value of actuator group E

A value can be preset for each actuator group A..E in this parameter. If a scene has been stored, the current object values of actuator groups A..E are overwritten with the values set here after programming or bus voltage failure and when the scene has been recalled again.

3.2.6.2 Parameters for control via *8-bit-scene*

This parameter window is visible, if a scene is controlled via *8-bit scene*.



Connected contact type

normally open: The input is closed in state of operation.
normally closed: The input is open in state of operation.

Control scene via

It is possible to select whether the scene control is carried out via *5 separate objects* or whether values that are stored in the actuators are recalled and saved via an *8 bit scene*. The parameters of the scene control via 5 separate objects is described in the previous section.

How does the scene control via *8-bit scene* function?

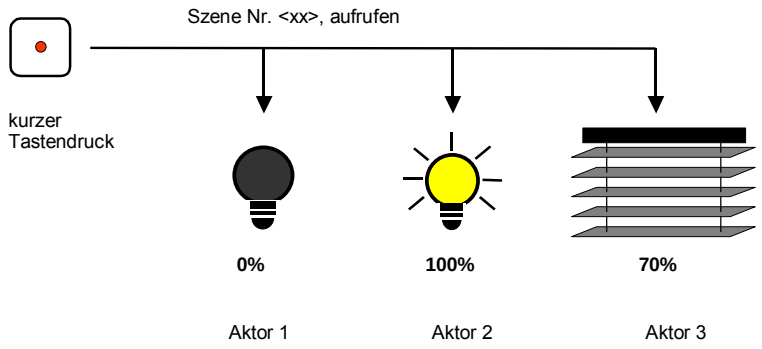


Diagram 3: Recalling and storing via 8-bit-scene

In the 8-bit scene, the push button gives the actuator the instruction to recall a scene. The scene is stored not in the push button but in the actuator. All the actuators are addressed via the same group address. A single telegram is sufficient to recall the scene.

The telegram value contains a scene number which must match the scene number in the parameters of the actuator. Up to 64 different scenes are managed via a single group address.

An 8-bit scene telegram contains the following information:
– Number of the scene (1...64)
– Recall scene / store scene

After a long push button action, the device sends a save command which causes the actuators to store the currently issued value as a new scene value.

What is the difference to a scene via *5 separate objects*?

The main difference is that the 8-bit scene is stored in the actuators. This requires actuators which support this function.

How does the user operate a scene?

The operation is very simple: A short operation calls a scene. A long operation stores a scene.

Number of scene

The scene number (1...64) is defined here.

Reaction on short operation

This parameter indicates whether a lightscene is recalled after a short operation of the input or whether no reaction should take place.

Store scene

This parameter defines in which way the storing of the current scene can be triggered as well as the functionality of the object “Store scene”. This is dependent on the scene control. The following table provides an overview:

Parameter value	Behaviour
“on long operation”	On a long keypress the object 8-bit scene sends a storage command on the bus. This triggers the actuator to store of the current scene. The object *Store scene* has no function.
“if object value = 1“	The object “8-bit scene” will send a storage command, if the object *Store scene* receives the value “1”.

“on long operation (if object value = 1)”	A long keypress triggers a storage command via object *8-bit-scene* only if the object *store scene* has the value “1“. A long operation will be interpreted like a short operation if, the object value *Store scene* has the value “0” or if no “1” has been received on this object since the last long keypress. Normally, the scene will thus be called.

Long operation after

This parameter is visible if the storing of the scene is possible via a long operation. The period after which an operation is interpreted as “long” is defined here.

Debounce time

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 4.1 for the exact function of this parameter.

3.2.6.3 Communication objects “Scene control”

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 Bit EIS1 DPT 1.003**	**CW**
0: Release channel 1: Disable channel The channel circuitry can be blocked or released via this communication object. A blocked channel behaves as if the input signal does not occur. The communication objects of the channel continue to be available. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the status of the input has changed during the blocking. If the channel is just being operated as it is being enabled, the channel behaves as if the operation has just commenced. The behaviour of the channel is undefined if the channel is blocked during operation.				
1 ... 5	**Actuator group A switch ... Actuator group E switch**	**Channel A ... Channel A**	**1 Bit EIS1 DPT 1.001**	**CWT**
1 ... 5	**Actuator group A value ... Actuator group E value**	**Channel A ... Channel A**	**1 Byte EIS6 DPT 5.010**	**CWT**
These objects are visible if the scene is controlled via *5 separate objects*. They control up to 5 actuator groups, either via 1 bit or 8 bit values (can be parameterised). When storing the scene, the device reads out the current value via the bus and stores it in these objects. On bus voltage recovery, the object values are overwritten with the parameterised values.				
1	**8-bit-scene**	**Channel A**	**1 Byte, EIS6 DPT 18.001**	**CT**
This object is visible if the control is carried out via an *8 bit scene*. It sends a scene number and the information as to whether a scene should be recalled or the current scene should be stored. The storing of the scene is carried out in the actuator. Telegram code in bits: MxSSSSSS (MSB) (LSB) M: 0 – Scene is recalled 1 – Scene is stored x: Not used S: Number of scene (0...63 according scene no. 1...64) A table of the object values can be found in section 6.2.				

No.	Function	Object name	Data type	Flags
6	**Store scene**	**Channel A**	**1 Bit EIS1 DPT 1.003**	**CWT**

This object can be used to trigger the saving of a scene via the bus or to indicate that the scene has been stored. The function depends the parameter *Store scene.*

Function on receiving a telegram:
0: Complete storage of a scene
1: Start storage of a scene

Function on sending a telegram:
0: Storage of a scene was completed
1: Storage of a scene was started

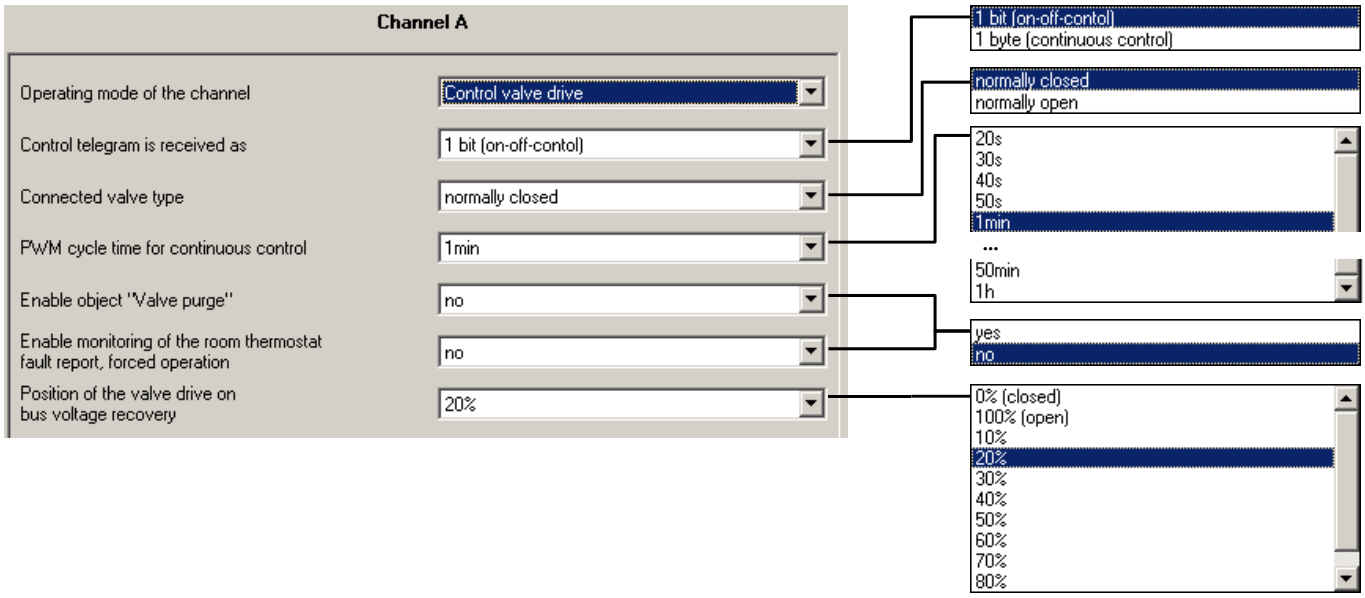
The object can be used to trigger or signal a scene storage

3.2.7 Operating mode: ”Control valve drive”

The “control valve drive operating mode” is described in the following.

This operating mode allows the control of an electronic relay, to which an electrothermal valve drive can be connected.

3.2.7.1 Parameters



Control telegram is received as

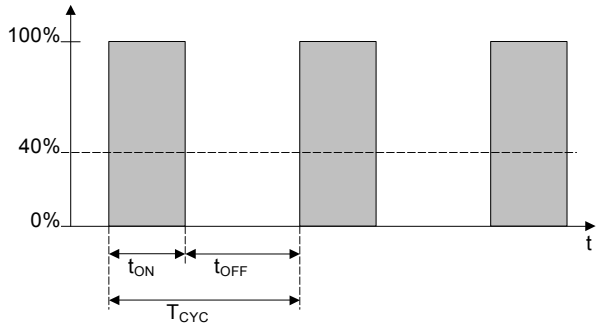
The heating actuator can either be controlled via the 1 bit object *Switch* or the 1 byte object *Control value (PWM)*.

In the case of **1 bit** control, the heating actuator functions in a similar way to a normal switch actuator: The room thermostat regulates the heating actuator via normal switching commands (*ON* and *OFF*). It is thus possible to implement a simple 2-step closed-loop control or a pulse width modulation of the control value.

In the case of **1 byte** control, a value of 0…255 (corresponds to 0%…100%) is preset by the room thermostat. This function is usually referred to as “continuous-action control”. The valve is closed at 0% and fully opened at 100%. The heating actuator controls intermediate values via pulse width modulation (see diagram below).

What does “Control via pulse width modulation (PWM)” mean?

If the room thermostat sends a 1-byte as a control value to the Universal Interface, the value will be transformed into a pulsed output signal (“PWM signal”). The control is carried out via a variable mark-to-space ratio. The following example clarifies this:



During t_{ON}, the valve is triggered with OPEN (“ON phase”). During t_{OFF}, the valve is triggered with CLOSE (“OFF phase”). See example: due to t_{ON} = 0.4 x T_{CYC}, the valve is set at approx. 40%. T_{CYC} is the so-called PWM cycle time for continuous control.

Connected valve type

In this parameter, it is possible to set whether a valve is “de-energised closed” or “de-energised opened”.

de-energised closed:
electronic relay is closed → valve is open.

de-energised open:
electronic relay is closed → valve is closed.

PWM cycle time for continuous control

When 1 byte control is selected, this parameter sets the PWM cycle time T_{CYC} which is used to time the control signal.

For 1 bit control and 1 byte control, this period is only used when the actuator is controlled in fault mode, during forced positioning and directly after bus voltage recovery.

Enable object “Valve purge”

The object *Valve purge* is enabled with this parameter.

Enable monitoring of the room thermostat, fault report, forced operation

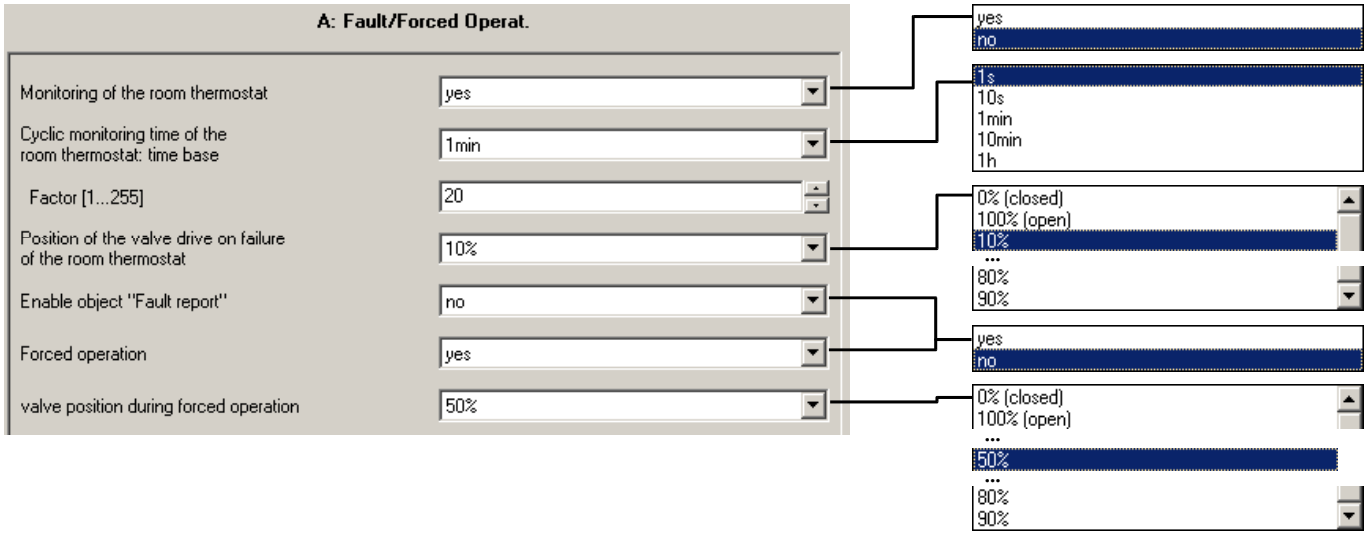
The parameter window *A: Fault-Forced Operat.* is enabled with this parameter. Further settings can be carried out in this window for the cyclical monitoring of the room thermostat and for the forced positioning of the actuator.

Position of the valve drive after bus voltage recovery

This parameter defines how the valve drive is controlled after bus voltage recovery, until the first switching or positioning command of the room thermostat is received. The position is controlled via a PWM signal.

3.2.7.2 Parameter window „Fault / Forced operat.“

This parameter window is visible if the value *yes* is entered in the parameter *Enable monitoring of the room thermostat, fault report, forced operation.*



Monitoring of the room thermostat

The cyclical monitoring of the room thermostat is enabled with this parameter.

The telegrams of the room thermostat are transmitted to the electronic actuator at specific cyclic intervals. If one or more of these telegram sequences is omitted, there may be a communications fault or a defect in the room thermostat.

If no telegrams are sent to the object *Switch* or *Control vale (PWM)* for the duration of the ***Cyclic monitoring time of the control value (time base x factor)***, the actuator switches to fault mode and triggers a safety position. The fault mode is finished as soon as a telegram is received again.

Position of the valve drive on failure of the room thermostat

This parameter defines the safety position which the actuator triggers in fault mode. The PWM cycle time T_{CYC} of the control is defined in the parameter *PWM cycle time for continuous control*.

Enable object “Fault report”

The object *Fault report* can be enabled in this parameter. It has the object value *ON* during fault mode. If there is no fault, it has the object value *OFF*. The object is always sent cyclically. The cyclic transmission time is identical to the cyclic monitoring time.

Forced operation

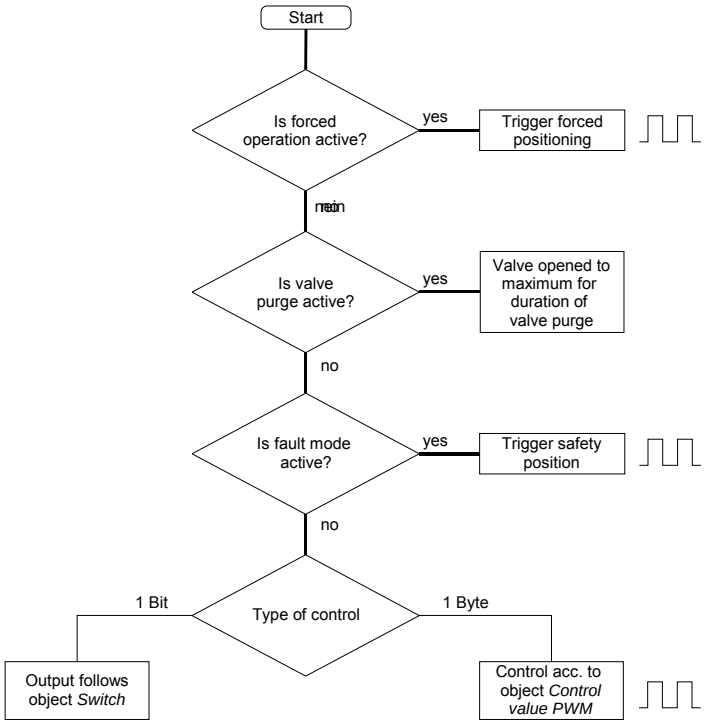
This parameter enables the forced operation function (object *Forced operation)*. During forced operation, the actuator triggers a freely selectable forced position. This has the highest priority i.e. it is not modified by a valve purge or a safety position.

Valve position during forced operation

In this parameter, the valve position triggered by the actuator is defined during the forced positioning. The PWM cycle time T_{CYC} of the control is defined in the parameter *PWM cycle time for continuous control*.

How about the priority between “Forced operation”, “Valve purge” and “Fault of room thermostat”?

The device can trigger specific special positions during “Forced positioning”, “Valve purge” and “Safety position”. The following diagram provides an overview:



The sequence in the table simultaneously indicates the priority of the special positions. Forced positioning has the highest priority.

How quick is the triggering of the special positions?

To improve the control behaviour, the special positions are sometimes not started or finished immediately but only once a PWM cycle time has elapsed or after an ON or OFF phase within the cycle.

The following table provides an overview:

Triggering of the valve via	**Behaviour at start**	**Behaviour at end**
Forced operation	Trigger immediately	Once an ON or OFF phase has elapsed.
Valve purge	Trigger immediately	Stop immediately
Fault mode	Once the cycle has elapsed	Once the cycle has elapsed

3.2.7.3 Communication objects „Control valve drive"

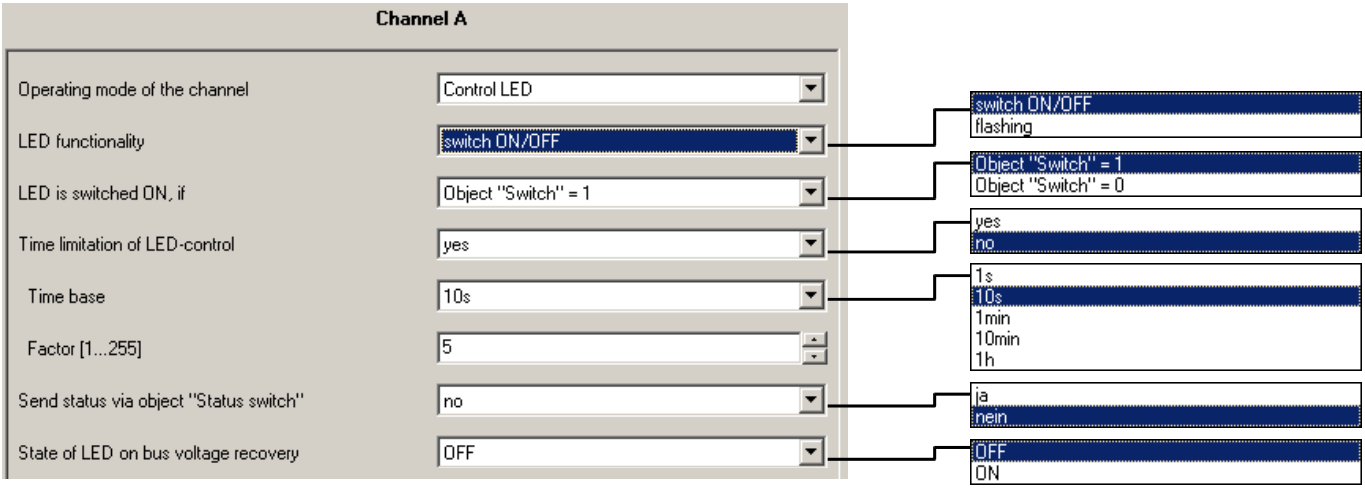
No.	Function	Object name	Data type	Flags
1	**Switch**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CW**
0: Close valve 1: Open valve This object is visible if the control of the heating actuator is carried out via a 1 bit object. If the object has the value "ON", the valve is opened while the valve is closed if the object has the value "OFF".				
1	**Control value (PWM)**	**Channel A**	**8 Bit EIS6 DPT 5.010**	**CW**
0: Close valve ... Mark-to-space ratio 255: Open valve This object is visible if the control of the heating actuator is carried out via an 8 bit object e.g. during continuous control. The object value (0…255) determines the selection ratio (mark-to-space ratio) of the valve.				
3	**Valve purge**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CW**
0: Stop valve purge 1: Start valve purge This object is visible if the parameter *Enable object 'Valve purge'* has the value *yes*. The valve purge of the device is activated or deactivated via this object. During the valve purge, the valve is controlled with "Open".				
4	**Forced operation**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CW**
0: Stop forced operation 1: Start forced operation This object is visible if 1 bit forced positioning is enabled in the parameters. The forced operation of the device is activated or deactivated via this object. In this way, the valve can be controlled with a defined value. Forced operation has the highest priority.				
5	**Status switch**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CT**
0: Valve is closed 1: Valve is opened This object reports the switching state of the heating actuator. The object value is sent after each change of the output. **Note:** **For PWM continuous control, this object is sent after each change in the output. The additional telegram load should therefore be taken into account!**				
6	**Fault report**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CT**
0: No fault 1: Fault mode active This object is visible if the fault message has been enabled in the parameters. If the output does not receive any telegrams from the room thermostat via the object *Switch* or *Control value (PWM)* for an adjustable period, the devices switches to fault mode and reports this via the object.				

3.2.8 Operating mode: ”Control LED“

The “control LED operating mode” is described in the following.

3.2.8.1 Parameters of LED function “switch ON/OFF”

Parameter window for LED function = *switch ON/OFF*



LED functionality

This parameter defines whether the output should control the LED permanently (*switch ON/OFF*) or whether it should flash. The corresponding objects *LED switch* or *LED flashing* are enabled.

LED is switched on, if

This parameter is visible if the LED function has been set to *switch ON/OFF*. It can be defined in which state of the object *LED switch* the LED is switched on.

Time limitation of LED-control

If *yes* has been entered in this parameter, the operating time of the LED has a time restriction.

Time base, Factor

If the time limit is active, it is possible to indicate in this parameter the maximum period that an LED is switched on. Once this time limit has elapsed, the LED is switched off automatically.

Period = time base x factor

Send status via object “Status switch”

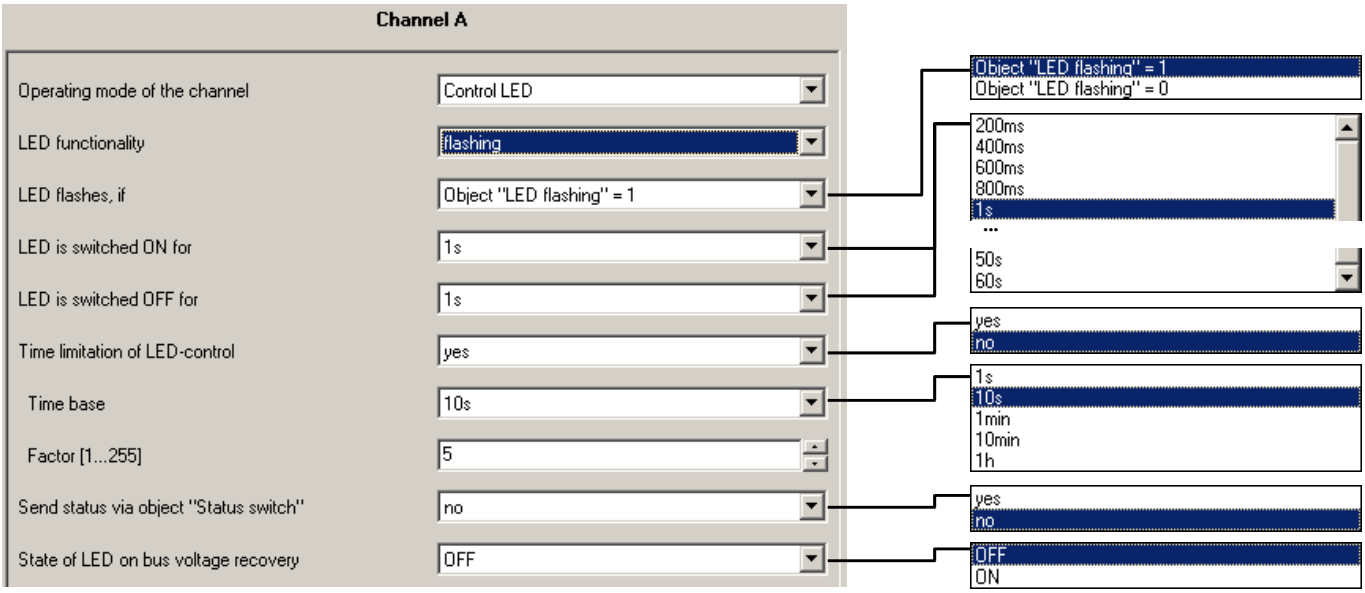
The object *Status switch* is enabled via this parameter. The object value is ON if the LED has been switched on.

State of LED after bus voltage recovery

It can be set whether the LED is switched ON or OFF after bus voltage recovery.

3.2.8.2 Parameters of LED function “flashing”

Parameter window for *LED functionality* = *Flashing*



LED flashes, if

This parameter is visible if the LED function *Flashing* has been set. It can be defined which state the object *LED flashing* must have so that the flashing of the LED is active.

LED is switched ON for
LED is switched OFF for

It is defined how long the LED is switched on or switched off during the flashing signal. The flash rate of the signal can thus be set.

Time limitation of LED control

If *yes* has been entered in this parameter, the flashing of the LED has a time restriction.

Time base, Factor

If the time limit is active, it is possible to indicate in this parameter the maximum period that an LED is flashing. Once this time limit has elapsed, the LED is switched off automatically.

Period = time base x factor

Send status via object “Status switch”

The object *Status switch* is enabled via this parameter. The object value is ON if the LED is flashing.

State of LED after bus voltage recovery

It can be set whether the LED is flashing (*ON*) or not (*OFF*) after bus voltage recovery.

3.2.8.3 Communication objects „Control LED“

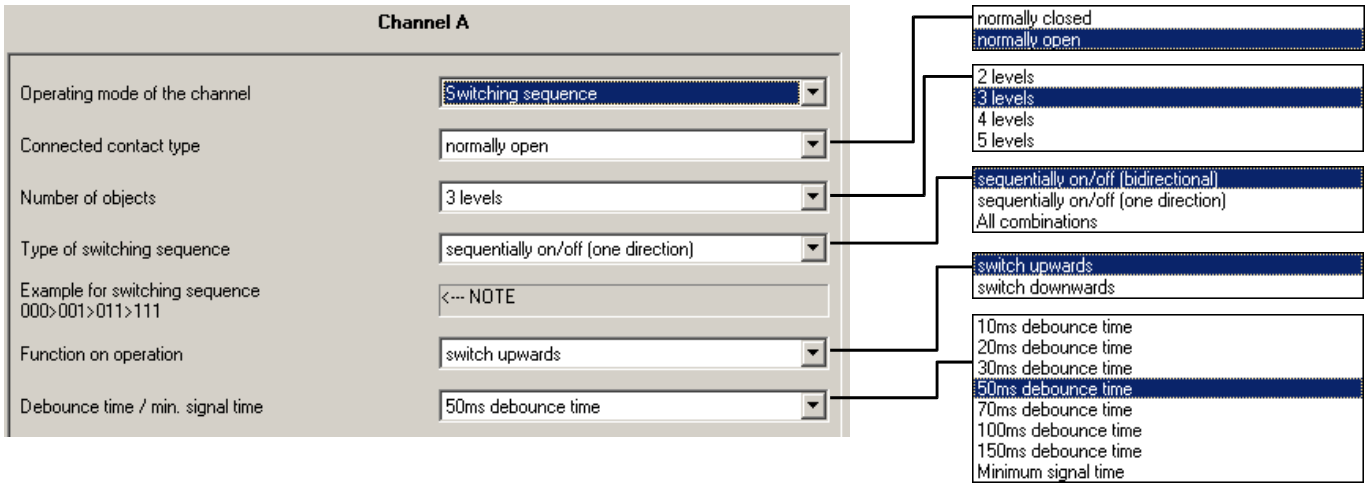
No.	Function	Object name	Data type	Flags
1	**LED switching**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CW**
This object is visible if the parameter “LED function” has been set to *switch ON/OFF*. The object switches the LED on and off. The telegram values can be set in the parameters.				
2	**LED flashing**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CW**
0: Stop flashing 1: Start flashing This object is visible if the parameter *LED function* has been set to *Flashing*. The flashing of the LED can be started and stopped via this object.				
3	**LED permanent ON**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CW**
0: Output follows the objects *LED switching* or *LED flashing* 1: LED permanently on The LED can be switched on permanently via this object. For example, the flashing function is deactivated in this way.				
4	**Status switch**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CT**
0: LED is switched off 1: LED is switched on or flashes This object is visible if the value *yes* has been set in the parameter *Send status via object ….*				

3.2.9 Operating mode: “Switching sequence“

The “switching sequence operating mode” is described in the following.

It enables up to five switch objects to be modified in a defined sequence by actuation of just a single input.

3.2.9.1 Parameters



Connected contact type

normally open: The input is closed in state of operation.
normally closed: The input is open in state of operation.

Number of objects

This parameter defines the number of objects used in the switching sequence. The communication objects “Value 1” to “Value 5” are enabled accordingly.

Type of switching sequence

The type of switching sequence can be selected here. The following switching sequences are available:

Switching sequence	Example (3 Objects)
“Sequentially on/off (bidirectional)”	...-000-001-011-111-011-001-...
“Sequentially on/off (one direction)”	000-001-011-111
All combinations	...-000-001-011-010-110-111-101-100-...

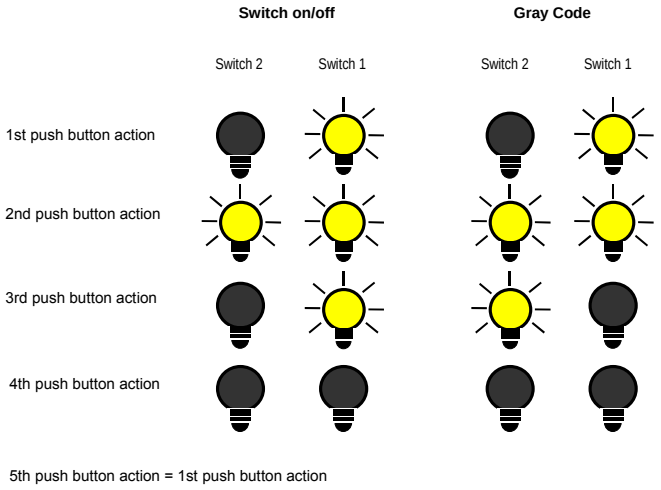
The sequence is switched one level further after each operation. The switching sequence relates to the states of three communication objects (0 = OFF, 1 = ON).

You can find a table of the sequence “All combinations” in the appendix 6.1.

What is the operating mode “switching sequence” used for?

The “Switching sequence” function enables up to five switch objects to be modified in a defined sequence by actuation of just a single input. The sequence is switched one level further after each operation.

Example:



In this example, two groups of luminaries are controlled. Thus two objects are used.

How many luminaries can be switched in a sequence?

Up to 5 luminaires (or groups of luminaries) can be switched.

Which switching sequences are available?

1. “Sequentially on/off (bidirectional)”

With this sequence a further communication object is switched on each actuation. When all the communication objects are switched on, they are switched off again one after the other, starting with the last object that was switched on.

2. “Sequentially on/off (one direction)

With this sequence a further communication object is switched on with each actuation. If all the communication objects are switched on, all further operations are ignored. At least two inputs are therefore required, one of which switches up a level and the other which switches down a level in the sequence.

3. „All combinations“

This switching sequence runs through all the combinations of the communication objects in succession. Only the value of one communication object is changed between two switching levels. A clear application of this switching sequence is e.g. the switching of two groups of luminaries in the sequence
00 – 01 – 11 – 10 – 00 ...

You can find a table of this sequence in the appendix 6.1.

How does the device know about the current position within the sequence?

The device determines the current position from the object values.

Is it possible to operate a switching sequence from several push-buttons?

Yes, the object “Level increment/decrement” can be used for this.

Example: Switching sequence “Sequentially on/off (bidirectional)” using three communication objects:

Switching level		**Value of the communication objects**		
No.	**Short code**	**“Switch 1”**	**“Switch 2”**	**“Switch 3”**
0	000	OFF	OFF	OFF

1	001	OFF	OFF	ON
2	011	OFF	ON	ON
3	111	ON	ON	ON
4	011	OFF	ON	ON
5	001	OFF	OFF	ON
0	...			

Short Code: ...>000>001>011>111>011>001>...

Function on operation

Only visible in the switching sequence *sequentially on/off (one direction)*. It can be set whether an operation of the push button switches up or down a level.

In this switching sequence at least two push buttons are necessary: One push button switches upwards, the other switches downwards.

Debounce time / minimum signal time

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 4.1 for the exact function of this parameter.

3.2.9.2 Communication objects „Switching sequence“

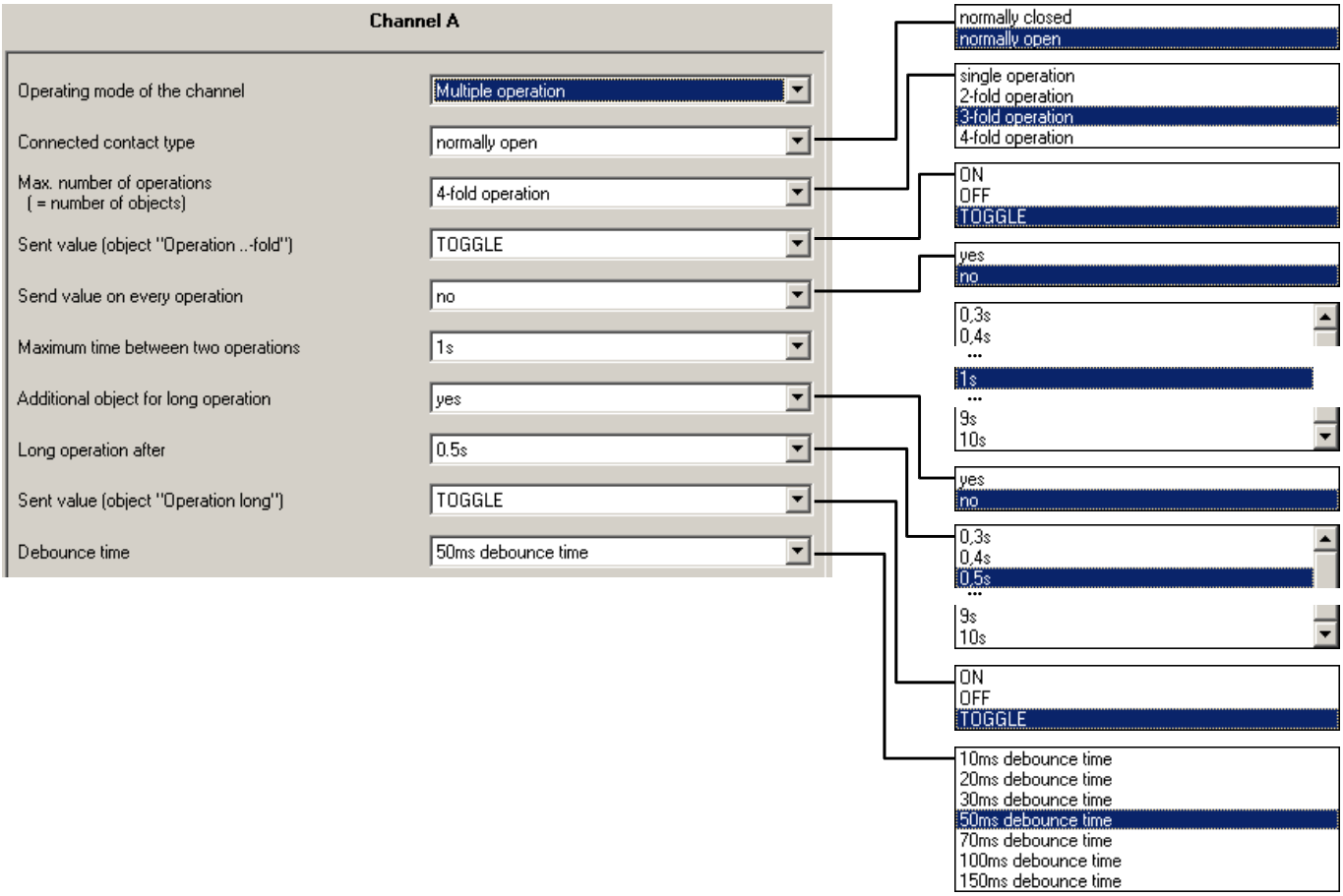
No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 Bit EIS1 DPT 1.003**	**CW**
0: Release channel 1: Disable channel The channel circuitry can be blocked or released via this communication object. A blocked channel behaves as if the input signal does not occur. The communication objects of the channel continue to be available. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the status of the input has changed during the blocking. If the channel is just being operated as it is being enabled, the channel behaves as if the operation has just commenced. The behaviour of the channel is undefined if the channel is blocked during operation.				
1 **...** **5**	**Switch 1** **...** **Switch 5**	**Channel A** **...** **Channel A**	**1 Bit EIS1 DPT 1.001**	**CT**
The number of these objects (max. 5) is set in the parameter *Number of objects*. The objects represent the values within a switching sequence.				
6	**Level increment/decrement**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CW**
0: Switch up one level 1: Switch down one level On receipt of an ON telegram at this communication object, the input switches up one level in the switching sequence. On receipt of an OFF telegram, it switches down one level.				

3.2.10 Operating mode: “Multiple operation”

The “multiple operation operating mode” is described in the following.

This operating mode enables the detectionof multiple operation within a certain period and to carry out defined switching actions accordingly.

3.2.10.1 Parameters



Connected contact type

normally open: The input is closed in state of operation.
normally closed: The input is open in state of operation.

Max. number of operations

This parameter specifies the maximum permitted number of operations. This number is identical to the number of communication objects *x-fold operation*.

Note: If the actual number of operations is higher than the maximum value set here, the input reacts as if the number of operations were identical to the maximum value set here.

Sent value (object “…-fold operation”)

It can be set here which object value should be sent. The settings *ON*, *OFF* and *TOGGLE* are possible. The current object value is inverted in the *TOGGLE* setting.

Send value on every operation

If *yes* is entered in this parameter, the associated object value is updated and sent after each operation in the case of multiple push button actions.

Example: For three-fold operations, the objects *1-fold operation* (after the first operation), *2-fold operation* (after the second operation) and *3-fold operation* (after the third operation) are sent.

Maximum time between two operations

This parameter sets the maximum interval between two operations.

After an operation, there is a delay for the duration specified here. If there are no further operations within this period, the object *…-fold operation* is sent and the period restarts after the next operation.

Additional object for long operation

After a long operation of the input, a further function can be executed via the object *Long operation*. If a long operation is carried out after one or several short operations within the maximum period, the short operations are ignored.

Long operation after

The period is defined here, after which an operation is interpreted as “long”.

Sent value (object “Long operation”)

It can be set here whether the object value *Long operation* should be switched on, switched off or toggled after a long operation.

Debounce time

The debounce prevents unwanted multiple operation of the input e.g. by bouncing of the contacts. Refer to section 4.1 for the exact function of this parameter.

3.2.10.2 Communication objects „Multiple operation“

No.	Function	Object name	Data type	Flags
0	**Blocking**	**Channel A**	**1 Bit EIS1 DPT 1.003**	**CW**
0: Release channel 1: Disable channel The channel circuitry can be blocked or released via this communication object. A blocked channel behaves as if the input signal does not occur. The communication objects of the channel continue to be available. When a disabled input is enabled, no telegrams are initially sent on the bus, even if the status of the input has changed during the blocking. If the channel is just being operated as it is being enabled, the channel behaves as if the operation has just commenced. The behaviour of the channel is undefined if the channel is blocked during operation.				
1 **...** **4**	**1-fold operation** **...** **4-fold operation**	**Channel A** **...** **Channel A**	**1 Bit EIS1 DPT 1.001**	**CT**
The number of these objects (max. 4) is set in the parameter *Max. number of operations*. After multiple operation of an input, the corresponding object is sent according to the number of operations. The telegram value can be set in the parameters.				
5	**Long operation**	**Channel A**	**1 Bit EIS1 DPT 1.001**	**CT**
This object is visible if the value *yes* has been set in the parameter *Additional object for long operation*. This object is sent once a long operation has been detected. The telegram value can be set in the parameters.				

3.3 Programming

The device can be programmed via the EIB Tool Software ETS2 **V1.2a** or higher. To reduce the programming time of the device via ETS, it is supplied as pre-programmed. During the programming, it is automatically detected whether the device already has the correct application program.

If the device has been pre-programmed with another version, which should only happen in exceptional cases, a full download is required. This can take several minutes.

Note: If a programmed application program needs to be reprogrammed, the device must first be unloaded via the ETS. This can be necessary in rare cases e.g. if an error occurs during a download.

Important: If a device has no function after programming, please make a new product import of the Universal Interface (.VD2-file) into the ETS and repeat the programming.

4 Special functions

The following section outlines special functions which could not be described in connection with the parameters.

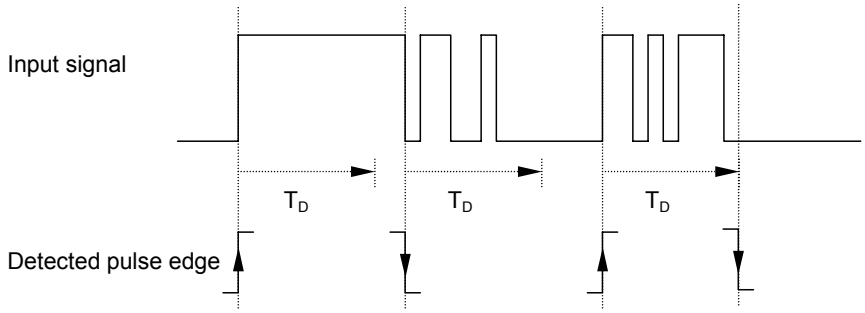
4.1 Debounce time an minimum signal time

A debounce time or a minimum operation time can be defined for each input.

Debounce time

If a pulse edge is detected at the input, the input reacts to it immediately (e.g. with the sending of a telegram). The debounce time T_D starts at the same time. During the debounce time, the signal is not evaluated at the input.

The following example clarifies this:



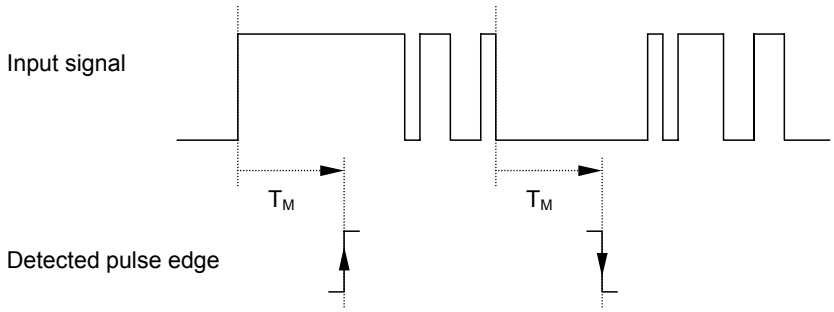
When a pulse edge is detected at the input, further pulse edges are ignored for the duration of the debounce time T_D.

Minimum signal time

This function differs from the debounce time in that the telegram is only sent once the minimum operation time has elapsed. The function operates as follows:

If a pulse edge is detected at the input, the minimum operation time starts. No telegrams are sent on the bus at this point. The signal at the input is observed during the minimum operation time. If a further pulse edge occurs at the input during this period, it is interpreted as a new operation and the minimum operation time is restarted. If the input signal does not change during the minimum operation time, a pulse edge is detected and a telegram is likewise sent on the bus.

The following example clarifies this:



Since only two pulse edges remain stable for the duration of the minimum operation time T_M, only these edges are recognised as valid.

4.2 Behaviour on bus voltage failure

After bus voltage failure, the device switches to energy saving mode for a short period in order to retain the stored values for as long as possible. If the bus voltage recovers during energy saving mode, the status of the device is fully maintained.

After a bus voltage failure of approx. 300 ms (duration is dependent on the function of the device), the energy saving mode is completed and the temporary memory is deleted. All the object values are identical to “0” and the device carries out an initialisation after bus voltage recovery.

4.3 Behaviour on bus voltage recovery

The behaviour depends on the operating mode where it can be set in the parameters in most cases. A detailed description can be found in section 3.2.1.1.

6 Appendix

6.1 Switching sequence: “All combinations”

The sequence is characterised by the fact that only one value changes between two levels. The transition to the next level therefore only requires the sending of a single telegram.

The following table describes the grey code when 5 objects are used:

Switching level		Value of the communication objects				
No.	**Short code**	**“Switch 5”**	**“Switch 4”**	**“Switch 3”**	**“Switch 2”**	**“Switch 1”**
0	00000	OFF	OFF	OFF	OFF	OFF
1	00001	OFF	OFF	OFF	OFF	ON
2	00011	OFF	OFF	OFF	ON	ON
3	00010	OFF	OFF	OFF	ON	OFF
4	00110	OFF	OFF	ON	ON	OFF
5	00111	OFF	OFF	ON	ON	ON
6	00101	OFF	OFF	ON	OFF	ON
7	00100	OFF	OFF	ON	OFF	OFF
8	01100	OFF	ON	ON	OFF	OFF
9	01101	OFF	ON	ON	OFF	ON
10	01111	OFF	ON	ON	ON	ON
11	01110	OFF	ON	ON	ON	OFF
12	01010	OFF	ON	OFF	ON	OFF
13	01011	OFF	ON	OFF	ON	ON
14	01001	OFF	ON	OFF	OFF	ON
15	01000	OFF	ON	OFF	OFF	OFF
16	11000	ON	ON	OFF	OFF	OFF
17	11001	ON	ON	OFF	OFF	ON
18	11011	ON	ON	OFF	ON	ON
19	11010	ON	ON	OFF	ON	OFF
20	11110	ON	ON	ON	ON	OFF
21	11111	ON	ON	ON	ON	ON
22	11101	ON	ON	ON	OFF	ON
23	11100	ON	ON	ON	OFF	OFF
24	10100	ON	OFF	ON	OFF	OFF
25	10101	ON	OFF	ON	OFF	ON
26	10111	ON	OFF	ON	ON	ON
27	10110	ON	OFF	ON	ON	OFF
28	10010	ON	OFF	OFF	ON	OFF
29	10011	ON	OFF	OFF	ON	ON
30	10001	ON	OFF	OFF	OFF	ON
31	10000	ON	OFF	OFF	OFF	OFF

6.2 Value table of object “8-bit-scene”

Object value		Meaning
decimal	**hexadecimal**	
00 *or* 64 01 *or* 65 02 *or* 66 ... 63 *or* 127	00h *or* 40h 01h *or* 41h 02h *or* 42h ... 3Fh *or* 7Fh	Call scene 1 Call scene 2 Call scene 3 ... Call scene 64
128 *or* 192 129 *or* 193 130 *or* 194 ... 191 *or* 255	80h *or* B0h 81h *or* B1h 82h *or* B2h ... AFh *or* FFh	Store scene 1 Store scene 2 Store scene 3 ... Store scene 64

6.3 Ordering information

Description	Ordering Info. Short code	Product code	bbn 40 16779 EAN	Unit price [EURO]	Price group	Weight 1 Pc. [kg]	Package [St.]
Universal Interface, 12-fold, FM	US/U 12.2	2CDG 110 065 R0011	65012 0		26	0,05	1

ABB STOTZ-KONTAKT GmbH
Postfach 10 16 80, D-69006 Heidelberg
Tel (06221) 701-607
Fax (06221) 701-724
www.abb.de/stotz-kontakt

Technische Hotline:
Telefon (06221) 701-434
E-mail: eib.hotline@de.abb.com

2005-09-16